



# Red Hat

## Red Hat Software Certification 2025

### Red Hat Software Certification Workflow Guide

For Use with Red Hat Enterprise Linux and Red Hat OpenShift



# Red Hat Software Certification 2025 Red Hat Software Certification Workflow Guide

---

For Use with Red Hat Enterprise Linux and Red Hat OpenShift

## Legal Notice

Copyright © Red Hat.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

The Red Hat Software Certification Workflow Guide provides an overview of the certification process for Red Hat Partners who want to deploy their own applications, management applications or software on Red Hat OpenShift Platform utilizing operators in a jointly supported customer environment. Version 9.28 updated December 10, 2025.

## Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE .....	9
CHAPTER 1. INTRODUCTION TO RED HAT SOFTWARE CERTIFICATION PROGRAM .....	10
1.1. THE RED HAT CERTIFICATION PROGRAM OVERVIEW	10
1.2. CERTIFICATION WORKFLOW	11
1.3. GETTING HELP AND GIVING FEEDBACK	11
CHAPTER 2. ONBOARDING CERTIFICATION PARTNERS .....	13
2.1. ONBOARDING EXISTING CERTIFICATION PARTNERS	13
2.2. ONBOARDING NEW CERTIFICATION PARTNERS	13
2.3. EXPLORING THE PARTNER LANDING PAGE	14
CHAPTER 3. GENERATING API KEYS TO AUTOMATE THE FUNCTIONAL CERTIFICATION PROCESSES	16
PART I. CERTIFYING OR VALIDATING STANDALONE APPLICATIONS .....	18
CHAPTER 4. INTRODUCTION TO NON-CONTAINERIZED PRODUCT CERTIFICATION .....	19
CHAPTER 5. CERTIFICATION WORKFLOW FOR NON-CONTAINERIZED APPLICATION .....	20
5.1. CERTIFICATION ONBOARDING	20
5.2. CERTIFICATION TESTING	20
5.3. PUBLISHING THE CERTIFIED APPLICATION	21
CHAPTER 6. CREATING A PRODUCT .....	22
6.1. OVERVIEW	23
6.1.1. Complete product listing details	23
6.1.2. Complete company profile information	23
6.1.3. Certify your product	23
6.1.4. Add at least one product component	23
6.1.5. Certify components for your listing	24
6.2. PRODUCT INFORMATION	25
6.3. COMPONENTS	26
6.4. SUPPORT	28
6.5. REMOVING A PRODUCT	28
CHAPTER 7. ADDING CERTIFICATION COMPONENTS .....	29
7.1. CERTIFICATION	29
7.2. COMPONENT DETAILS	29
7.3. CONTACT INFORMATION	30
7.4. ASSOCIATED PRODUCTS	30
CHAPTER 8. SETTING UP THE TEST ENVIRONMENT FOR NON-CONTAINERIZED APPLICATION TESTING .....	31
8.1. SETTING UP A SYSTEM THAT ACTS AS A SYSTEM UNDER TEST	31
CHAPTER 9. DOWNLOADING THE TEST PLAN FROM RED HAT CERTIFICATION PORTAL .....	33
CHAPTER 10. RUNNING CERTIFICATION TESTS BY USING CLI AND DOWNLOADING THE RESULTS FILE .....	34
10.1. USING THE TEST PLAN TO PREPARE THE HOST UNDER TEST FOR TESTING	34
10.2. RUNNING THE CERTIFICATION TESTS USING CLI	34
10.3. REVIEWING AND DOWNLOADING THE RESULTS FILE OF THE EXECUTED TEST PLAN	35
CHAPTER 11. UPLOADING THE RESULTS FILE OF THE EXECUTED TEST PLAN TO RED HAT CERTIFICATION PORTAL .....	36

<b>CHAPTER 12. RECERTIFICATION .....</b>	<b>37</b>
<b>CHAPTER 13. PUBLISHING THE CERTIFIED APPLICATION .....</b>	<b>38</b>
<b>APPENDIX A. RUNNING THE CERTIFICATION TESTS BY USING COCKPIT .....</b>	<b>39</b>
A.1. CONFIGURING THE SYSTEM AND RUNNING TESTS BY USING COCKPIT	39
A.1.1. Setting up the Cockpit server	39
A.1.2. Adding system under test to Cockpit	40
A.1.3. Getting authorization on the Red Hat SSO network	40
A.1.4. Downloading test plans in Cockpit from Red Hat certification portal	41
A.1.5. Using the test plan to prepare the system under test for testing	41
A.1.6. Running the certification tests using Cockpit	42
A.1.7. Reviewing and downloading the results file of the executed test plan	42
A.1.8. Submitting the test results from Cockpit to the Red Hat Certification Portal	43
<b>PART II. CERTIFYING OR VALIDATING CONTAINERIZED APPLICATIONS .....</b>	<b>44</b>
<b>CHAPTER 14. WORKING WITH CONTAINERS .....</b>	<b>45</b>
14.1. INTRODUCTION TO CONTAINERS	45
14.2. CONTAINER CERTIFICATION WORKFLOW	46
14.2.1. Certification on-boarding	46
14.2.2. Certification testing for containerized applications	47
14.2.3. Publishing the certified product listing on the Red Hat Ecosystem Catalog	47
14.3. TESTING MULTI-ARCH CONTAINER CERTIFICATION USING PREFLIGHT	48
14.3.1. Building and pushing multi-arch container images using Podman	48
<b>CHAPTER 15. CREATE A PRODUCT .....</b>	<b>50</b>
15.1. OVERVIEW	51
15.1.1. Complete product listing details for containerized applications	51
15.1.2. Complete company profile information for containerized applications	51
15.1.3. Accept legal agreements for containerized applications	51
15.1.4. Certify your containerized applications	52
15.1.5. Add at least one product component for containerized applications	52
15.1.6. Certify components for your listing for containerized applications	52
15.2. PRODUCT INFORMATION	53
15.3. COMPONENTS TAB FOR CONTAINERS	55
15.3.1. Container images	55
15.3.2. Containerized application for RHEL	57
15.3.3. Containerized application for OpenStack	58
15.4. SUPPORT	59
15.5. REMOVING A PRODUCT	59
<b>CHAPTER 16. ADDING CERTIFICATION COMPONENTS .....</b>	<b>60</b>
16.1. IMAGES	60
16.2. CERTIFICATION FOR CONTAINERS	61
16.2.1. For Container images	61
16.2.2. For Containerized applications on RHEL	62
16.2.3. For Containerized applications for OpenStack	63
16.3. SECURITY	64
16.4. REPOSITORY INFORMATION	64
16.5. COMPONENT DETAILS	65
16.6. CONTACT INFORMATION	66
16.7. ASSOCIATED PRODUCTS FOR CONTAINERS	67
<b>CHAPTER 17. RUNNING THE CERTIFICATION TEST SUITE .....</b>	<b>68</b>

17.1. ALL CONTAINERIZED APPLICATIONS	68
17.2. ADDITIONAL STEPS FOR RHEL CONTAINERIZED APPLICATIONS	69
<b>CHAPTER 18. PUBLISHING THE CERTIFIED CONTAINER ON RED HAT ECOSYSTEM CATALOG .....</b>	<b>70</b>
18.1. PUBLISHING CERTIFIED CONTAINERS	70
<b>PART III. CERTIFYING OR VALIDATING OPERATORS .....</b>	<b>71</b>
<b>CHAPTER 19. WORKING WITH OPERATORS .....</b>	<b>72</b>
19.1. INTRODUCTION TO OPERATORS	72
19.2. CERTIFICATION WORKFLOW FOR OPERATORS	73
19.2.1. Certification on-boarding for Operators	73
19.2.2. Certification testing for Operators	73
19.2.3. Publishing the certified Operator on the Red Hat Ecosystem Catalog	74
<b>CHAPTER 20. CREATE A PRODUCT .....</b>	<b>75</b>
20.1. OVERVIEW	76
20.1.1. Complete product listing details for Operators	76
20.1.2. Complete company profile information for Operators	76
20.1.3. Accept legal agreements for Operators	76
20.1.4. Certify your Operators	76
20.1.5. Add at least one product component for Operators	77
20.1.6. Certify components for your listing for Operators	77
20.2. PRODUCT INFORMATION	78
20.3. COMPONENTS	80
20.4. SUPPORT	81
20.5. REMOVING A PRODUCT	82
<b>CHAPTER 21. ADDING CERTIFICATION COMPONENTS .....</b>	<b>83</b>
21.1. CERTIFICATION FOR OPERATORS	83
21.2. OPTIONAL QUALIFICATIONS FOR OPERATORS	84
21.3. REPOSITORY INFORMATION FOR OPERATORS	85
21.4. COMPONENT DETAILS FOR OPERATORS	85
21.5. CONTACT INFORMATION FOR OPERATORS	86
21.6. ASSOCIATED PRODUCTS FOR OPERATORS	86
21.7. UPDATE GRAPH	87
<b>CHAPTER 22. RUNNING THE CERTIFICATION SUITE WITH RED HAT HOSTED PIPELINE .....</b>	<b>88</b>
22.1. FORKING THE REPOSITORY	89
22.2. ADDING YOUR OPERATOR BUNDLE	89
22.2.1. If you have certified this operator before	89
22.2.2. If you are newly certifying this operator	90
22.3. FILE-BASED CATALOG	91
22.3.1. Onboarding	92
22.3.2. Submitting FBC changes	93
22.4. FILE-BASED CATALOG WORKFLOW	94
22.4.1. Enabling FBC in the workflow	94
22.4.2. FBC templates	94
22.4.3. Mapping templates to OpenShift catalogs	94
22.4.4. Generating catalogs from templates	95
22.4.5. Adding a new bundle to the catalog	96
22.4.5.1. Using Automated Release	96
22.4.5.2. Manually adding bundle	96
22.4.6. Updating existing catalogs	97
22.5. FILE-BASED CATALOG AUTO-RELEASE	97

22.5.1. Overview	97
22.5.2. Creating the release-config.yaml file	97
22.5.3. File structure details	98
22.6. CREATING A PULL REQUEST	99
22.6.1. Guidelines to follow	99
<b>CHAPTER 23. PUBLISHING THE CERTIFIED OPERATOR .....</b>	<b>100</b>
<b>CHAPTER 24. TROUBLESHOOTING GUIDELINES .....</b>	<b>101</b>
<b>APPENDIX B. RUNNING THE CERTIFICATION TEST SUITE LOCALLY .....</b>	<b>102</b>
B.1. ADDING YOUR OPERATOR BUNDLE	103
B.1.1. If you have certified this operator before -	103
B.1.2. If you are newly certifying this operator -	103
B.2. FORKING THE REPOSITORY	105
B.3. INSTALLING THE OPENSHIFT OPERATOR PIPELINE	106
B.3.1. Automated process	106
B.3.1.1. Prerequisites	106
B.3.1.2. Installing the pipeline through an Operator	108
B.3.1.3. Executing the pipeline	110
B.3.2. Manual process	110
B.3.2.1. Installing the OpenShift Pipeline Operator	110
B.3.2.2. Configuring the OpenShift (oc) CLI tool	110
B.3.2.3. Creating an OpenShift Project	111
B.3.2.4. Adding the kubeconfig secret	111
B.3.2.5. Importing Operator from Red Hat Catalog	111
B.3.2.6. Installing the certification pipeline dependencies	112
B.3.2.7. Configuring the repository for submitting the certification results	112
B.3.2.7.1. Adding GitHub API Token	112
B.3.2.7.2. Adding Red Hat Container API access key	112
B.3.2.7.3. Enabling digest pinning	112
B.3.2.7.4. Using a private container registry	113
B.4. EXECUTE THE OPENSHIFT OPERATOR PIPELINE	113
B.4.1. Running the Minimal pipeline	114
B.4.2. Running the pipeline with image digest pinning	115
B.4.3. Running the pipeline with a private container registry	115
B.5. SUBMIT CERTIFICATION RESULTS	116
B.5.1. Submitting test results from the minimal pipeline	116
B.5.2. Submitting test results with image digest pinning	117
B.5.3. Submitting test results from a private container registry	118
B.5.4. Submitting test results with image digest pinning and from a private container registry	118
<b>APPENDIX C. HELM AND ANSIBLE OPERATORS .....</b>	<b>120</b>
<b>PART IV. CERTIFYING OR VALIDATING HELM CHARTS .....</b>	<b>121</b>
<b>CHAPTER 25. WORKING WITH HELM CHARTS .....</b>	<b>122</b>
25.1. INTRODUCTION TO HELM CHARTS	122
25.2. CERTIFICATION WORKFLOW FOR HELM CHARTS	123
25.2.1. Certification on-boarding for Helm charts	123
25.2.2. Certification testing for Helm charts	123
25.2.3. Publishing the certified Helm chart on the Red Hat Ecosystem Catalog	124
<b>CHAPTER 26. VERIFYING HELM CHARTS FOR CERTIFICATION .....</b>	<b>125</b>
26.1. PREPARING THE TEST ENVIRONMENT	125

26.2. RUNNING THE HELM CHART-VERIFIER TOOL	126
26.2.1. By using Podman or Docker	127
26.2.1.1. Configuring the timeout option	129
26.2.1.2. Saving the report	129
26.2.1.3. Configuring the error log	130
26.2.2. By using the binary file	130
<b>CHAPTER 27. CREATE A PRODUCT .....</b>	<b>132</b>
27.1. OVERVIEW FOR HELM CHARTS	133
27.1.1. Complete product listing details for Helm charts	133
27.1.2. Complete company profile information for Helm charts	133
27.1.3. Accept legal agreements for Helm charts	133
27.1.4. Certify your Helm charts	133
27.1.5. Add at least one product component for Helm charts	134
27.1.6. Certify components for your listing for Helm charts	134
27.2. PRODUCT INFORMATION FOR HELM CHARTS	135
27.3. COMPONENTS FOR HELM CHARTS	137
27.4. SUPPORT FOR HELM CHARTS	139
27.5. REMOVING A PRODUCT	139
<b>CHAPTER 28. ADDING CERTIFICATION COMPONENTS .....</b>	<b>140</b>
28.1. CERTIFICATION FOR HELM CHARTS	140
28.2. OPTIONAL QUALIFICATIONS FOR HELM CHARTS	140
28.3. REPOSITORY INFORMATION FOR HELM CHARTS	141
28.4. COMPONENT DETAILS FOR HELM CHARTS	141
28.5. CONTACT INFORMATION FOR HELM CHARTS	142
28.6. ASSOCIATED PRODUCTS FOR HELM CHARTS	142
<b>CHAPTER 29. SUBMITTING YOUR HELM CHART FOR CERTIFICATION .....</b>	<b>144</b>
29.1. SUBMITTING A HELM CHART WITHOUT THE CHART VERIFICATION REPORT	145
29.1.1. Chart as a tarball	145
29.1.2. Chart in a directory	145
29.2. SUBMITTING A CHART VERIFICATION REPORT WITHOUT THE HELM CHART	146
29.2.1. For submitting a signed report	146
29.2.2. For submitting a report for a signed chart	147
29.3. SUBMITTING A CHART VERIFICATION REPORT ALONG WITH THE HELM CHART	147
29.3.1. For submitting a signed report	147
29.3.2. For submitting a signed Helm chart	147
29.4. SUMMARY OF CERTIFICATION SUBMISSION OPTIONS	147
29.5. VERIFICATION STEPS	149
<b>CHAPTER 30. PUBLISHING THE CERTIFIED HELM CHART .....</b>	<b>150</b>
<b>PART V. FUNCTIONAL CERTIFICATION FOR OPENSHIFT BADGES: VIRTUALIZATION, BEST PRACTICES, CNF, CNI, CSI .....</b>	<b>151</b>
<b>CHAPTER 31. OPENSHIFT VIRTUALIZATION VALIDATION .....</b>	<b>152</b>
31.1. VALIDATING YOUR PRODUCTS WITH THE OPENSHIFT VIRTUALIZATION FEATURE	152
31.2. CREATING A PRODUCT	152
31.2.1. Overview for OpenShift Virtualization	153
31.2.1.1. Complete product listing details	153
31.2.1.2. Complete company profile information	153
31.2.1.3. Accept legal agreements	154
31.2.1.4. Add at least one product component	154
31.2.1.5. Certify components for your listing	155

31.2.2. Product Information for OpenShift Virtualization	155
31.2.3. Components for OpenShift Virtualization	157
31.2.4. Support for OpenShift Virtualization	159
31.2.5. Removing a product	159
31.3. ADDING CERTIFICATION COMPONENTS	159
31.3.1. Validation	160
31.3.2. Component details	160
31.3.3. Contact information	160
31.3.4. Associated products	160
31.4. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG	161
<b>CHAPTER 32. MEETS BEST PRACTICES .....</b>	<b>163</b>
32.1. MEETING BEST PRACTICES IN CLOUD NATIVE SOFTWARE CERTIFICATION	163
32.2. CERTIFICATION ONBOARDING	163
32.3. CREATING A PRODUCT	163
32.4. ADDING COMPONENTS	163
32.5. CERTIFICATION TESTING	163
32.6. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG	164
<b>CHAPTER 33. CNF CERTIFICATION AND VALIDATION .....</b>	<b>166</b>
33.1. WORKING WITH CLOUD-NATIVE NETWORK FUNCTION (CNF) CERTIFICATION	166
33.1.1. Introduction to Cloud-native Network Function	166
33.1.2. Certification workflow for CNF	166
33.1.2.1. Certification onboarding for cnf	167
33.1.2.2. Completing the product listing for cnf	167
33.1.2.3. Publishing the product listing on the Red Hat Ecosystem Catalog	167
33.2. CREATE A PRODUCT	168
33.2.1. Overview for CNF	169
33.2.1.1. Complete product listing details for CNF	169
33.2.1.2. Complete company profile information for CNF	169
33.2.1.3. Accept legal agreements for CNF	170
33.2.1.4. Add at least one product component for CNF	170
33.2.1.5. Certify components for your listing for CNF	170
33.2.2. Product Information for CNF	171
33.2.3. Components for CNF	173
33.2.3.1. Certify components for your listing	174
33.2.4. Support for CNF	174
33.2.5. Removing a product	175
33.3. ADDING CERTIFICATION COMPONENTS	175
33.3.1. Certification for CNF	175
33.3.1.1. Validate the functionality of your CNF on Red Hat OpenShift	175
33.3.1.2. Certify the functionality of your CNF on Red Hat OpenShift	176
33.3.2. Component details for CNF	178
33.3.3. Contact Information for CNF	178
33.3.4. Associated products for CNF	178
33.4. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG	179
33.5. RECERTIFYING A CNF PACKAGE	181
<b>CHAPTER 34. CNI CERTIFICATION .....</b>	<b>182</b>
34.1. WORKING WITH CONTAINER NETWORK INTERFACE (CNI) CERTIFICATION	182
34.1.1. Introduction to Container Network Interface	182
34.1.2. Certification workflow for CNI	182
34.1.2.1. Certification on-boarding for CNI	183
34.1.2.2. Certification testing for CNI	183

---

34.1.2.3. Publishing the product listing on the Red Hat Ecosystem Catalog	183
34.2. CREATING A PRODUCT	184
34.3. ADDING CERTIFICATION COMPONENTS	184
34.4. WORKING WITH THE OPENSHIFT OPERATOR PIPELINE	184
34.5. CONFIGURING YOUR TEST ENVIRONMENT FOR RUNNING THE CNI TESTS	184
34.6. RUNNING THE CNI TESTS	185
34.7. SUBMITTING YOUR CNI OPERATOR FOR CERTIFICATION	186
34.8. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG	186
<b>CHAPTER 35. CSI CERTIFICATION .....</b>	<b>188</b>
35.1. WORKING WITH CONTAINER STORAGE INTERFACE (CSI) CERTIFICATION	188
35.1.1. Introduction to Container Storage Interface	188
35.1.2. Certification workflow for CSI	188
35.1.2.1. Certification on-boarding for CSI	189
35.1.2.2. Certification testing for CSI	189
35.1.2.3. Publishing the product listing on the Red Hat Ecosystem Catalog	189
35.2. CREATING A PRODUCT	190
35.3. ADDING CERTIFICATION COMPONENTS	190
35.4. WORKING WITH THE OPENSHIFT OPERATOR PIPELINE	190
35.5. CONFIGURING YOUR TEST ENVIRONMENT FOR RUNNING THE CSI TESTS	190
35.6. ACCESSING THE CSI CERTIFICATION TESTS	191
35.7. SETTING UP THE CSI TEST PARAMETERS	191
35.8. RUNNING THE CSI CERTIFICATION TESTS	191
35.8.1. Running the CSI function test	191
35.8.2. Running the KubeVirt storage checkup test	192
35.8.2.1. Running a Kubevirt storage checkup test by using the web console	192
35.8.2.2. Running a KubeVirt storage checkup test by using the command line	193
35.9. SUBMITTING CSI TEST RESULTS	193
35.10. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG	193



## 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 five main elements:

- **Product listing:** A source of all the essential product information that potential customers look for before using your product.
- **Components:** It comprises the containers, operators, helm charts, and various other infrastructure services that are attached to the product listing. Additionally, it includes 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.
  - **Containerized applications:** It has the following product categories:
    - **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.
    - **Functional certification for OpenShift badges:**
      - **Cloud-native Network Functions (CNFs):** Partner Validated and Certified CNFs are attached to the product listings and are published on the Red Hat Ecosystem Catalog.
      - **Container Network Interface (CNI)** Certified CNIs are published on the Red Hat Ecosystem Catalog.
      - **Container Storage Interface (CSI)** Certified CSIs are published on the Red Hat Ecosystem Catalog.
      - **Meets Best Practices:** Workload follows Red Hat's best practices.
    - **Applications on OpenStack Infrastructure:** Non-containerized, containerized, and VNF applications are certified on OpenStack Infrastructure and 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.

This table summarizes the basic differences between a product listing and components:

Product listing	Component (Project)
Includes detailed information about your product.	The individual containers, operators, helm charts, and infrastructure services that you test, certify, and then add to the product listing.
Products are composed of one or more components.	Components are added to a product listing.
You add components to a product for proceeding with certification.	A component can be used in multiple products by adding it to each product listing.
A product can not be published without certified components.	Certified components are published as part of a product listing.

## 1.2. CERTIFICATION WORKFLOW

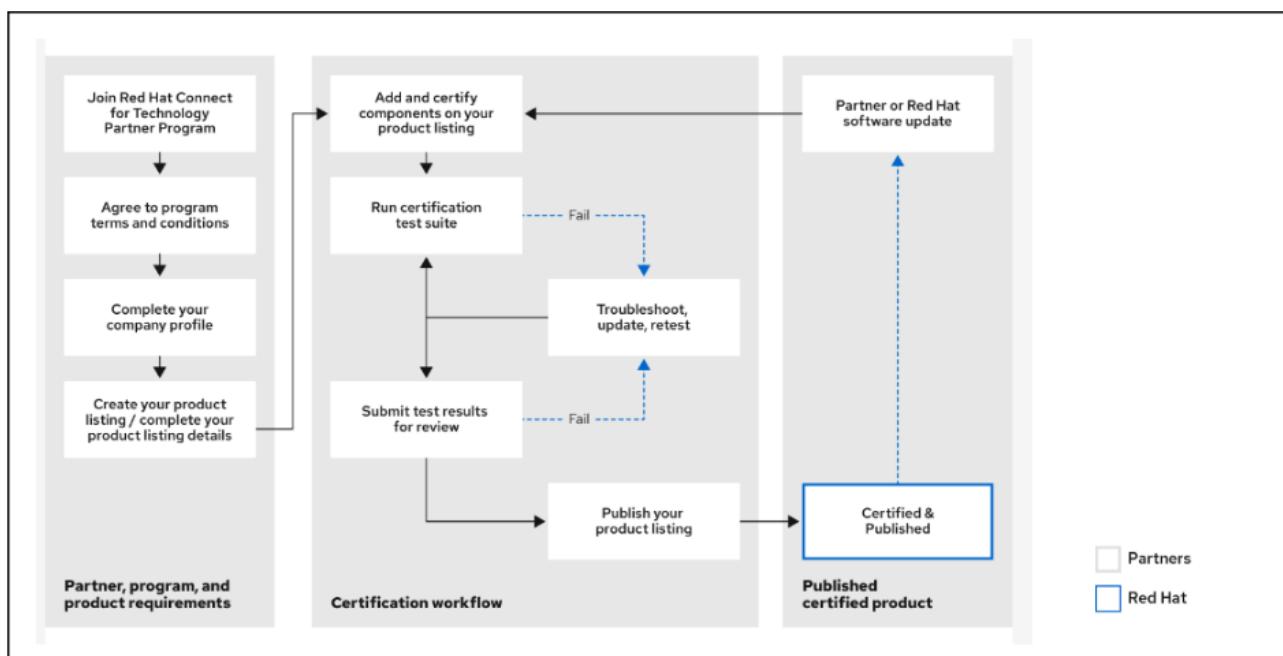


### NOTE

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 the software certification process.

Figure 1.1. Certification workflow



## 1.3. 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](mailto: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. 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
3. 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. Click **Register for a Red Hat account**

3. Enter the following details to create a new Red Hat account:

- a. 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. Select **Corporate** for 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.

e. Enter your **Contact information**.

f. 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.

## 2.3. EXPLORING THE PARTNER LANDING PAGE

After logging in to [Red Hat Partner Connect](#), the partner landing page opens. This page serves as a centralized hub, offering access to various partner services and capabilities that enable you to start working on opportunities.

The Partner landing page offers the following services:

- Certified technology portal
- Deal registrations
- Red Hat Partner Training Portal
- Access to our library of marketing, sales & technical content
- Help and support
- Email preference center
- Partner subscriptions
- User account

As part of the Red Hat partnership, partners receive access to various Red Hat systems and services that enable them to create shared value with Red Hat for our joint customers.

Select the **Certified technology portal** tile to begin your product certification journey. The personalized Certified Technology partner dashboard opens.

## CHAPTER 3. GENERATING API KEYS TO AUTOMATE THE FUNCTIONAL CERTIFICATION PROCESSES

A valid Pyxis API key is a token used to access partner-specific certification data through the REST API. With this key, you can directly access your certification components available in the Red Hat Partner Certification tool and seamlessly upload certification results without the need for repeated authentication.



### NOTE

API keys are tied to specific Organization IDs, allowing access only to the product listings associated with that particular organization.

### Prerequisites

- Have access to the [Red Hat Partner Certification](#) tool.
- A valid component ID for your certification components on the [Red Hat Partner Certification](#) tool.

### Procedure

Follow the procedure to generate the API keys:

1. After logging in to the [Red Hat Partner Certification](#) tool, go to the **Certification management** tab.
2. From the **Resources** listing, click **API keys**. The **API Keys** page opens.
3. Click **Add API key**. The **Add API Keys** dialog opens.
4. Enter a unique key in the **Name** text box.
5. Enter a brief description about the newly created key including its purpose in the **Description** text box.
6. Click **Confirm**.  
A new token is generated. The **Copy Token** wizard opens, allowing you to copy the generated API key.
7. Click the copy icon next to the generated API key to save it in a secured location.



### IMPORTANT

Red Hat doesn't store your API keys. If you don't save the key now, you can't retrieve it after closing this wizard.

### Verification

The new API key is listed on the **API Keys** page including the following details:

- Name
- Description

- User Name
- Created Time
- Actions - Click **Revoke**, to delete the key.

## PART I. CERTIFYING OR VALIDATING STANDALONE APPLICATIONS

## CHAPTER 4. 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.

**Partner Validation** - Select this type of certification, if you want to validate your product using your own criteria and test suite on Red Hat platforms. This partner validation allows you to publish your software offerings on the Red Hat Ecosystem Catalog more quickly. However, validated workloads may not incorporate all of Red Hat integration requirements and best practices. We encourage you to continue your efforts toward Red Hat certification.

**Certified** - Select this type of certification, if you want your product to undergo thorough testing by using Red Hat's test suite, and benefit from collaborative support. Your products will meet your standards and Red Hat's criteria, including interoperability, lifecycle management, security, and support requirements.

It is critical for partners to understand the distinction between Partner Validation and Red Hat Certification. Find below a clear differentiation between Validation and Certification programs:

Feature	Partner Validation	Red Hat Certification
Testing Criteria	Determined by Partner	Determined by both Red Hat and Partner
Testing Tools	Provided by Partner	Provided by both Red Hat and Partner
Red Hat Review	Partner statements, supportive documentation and test results	Test results, partner statements, and documentation
Scope	Quicker market entry, initial presence on the Red Hat Ecosystem Catalog	Comprehensive compliance with Red Hat platform requirements, best practices such as interoperability, security, and support

### Additional resources

- [Introduction to Partner Validation](#)

# CHAPTER 5. 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.

### Task Summary

The certification workflow includes three primary stages -

1. [Section 5.1, "Certification onboarding"](#)
2. [Section 5.2, "Certification testing"](#)
3. [Section 5.3, "Publishing the certified application"](#)

## 5.1. CERTIFICATION ONBOARDING

Perform the steps outlined for certification onboarding:

1. Join the [Red Hat Connect](#) for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Create your product listing by selecting your desired product category. You can select from the available product categories:
  - a. Containerized Application
  - b. Standalone Application
  - c. OpenStack Infrastructure
4. Complete your company profile.
5. Add components to the product listing.
6. Certify components for your product listing.

### Additional resources

For detailed instructions about creating your first product listing, see [Creating a product](#).

## 5.2. 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](#).

### 5.3. PUBLISHING THE CERTIFIED APPLICATION

The **Certified** application must be added 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.

## CHAPTER 6. CREATING A PRODUCT

The product listing provides marketing and technical information, showcasing your product's features and advantages to potential customers. It lays the foundation for adding all necessary components to your product for certification.

### 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 before certification.

### Procedure

Red Hat recommends completing all optional fields in the listing tabs for a comprehensive product listing. More information helps mutual customers make informed choices.

Red Hat encourages collaboration with your product manager, marketing representative, or other product experts when entering information for your product listing.

Fields marked with an asterisk (\*) are mandatory.

### Procedure

1. Log in to the [Red Hat Partner Connect Portal](#).
2. Go to the Certified technology portal tab and click **Visit the portal**.
3. On the header bar, click **Product management**.
4. From the **Listing and certification** tab click **Manage products**.
5. From the **My Products** page, click **Create Product**.  
A **Create New Product** dialog opens.
6. Enter the **Product name**.
7. From the **What kind of product would you like to certify?** drop-down, select the required product category and click **Create product**. For example, select **Standalone Application** for creating a non containerized product listing.

A new page with your Product name opens. It comprises the following tabs:

- [Section 6.1, "Overview"](#)
- [Section 6.2, "Product Information"](#)
- [Section 6.3, "Components"](#)
- [Section 6.4, "Support"](#)

Along with the following tabs, the page header provides the **Product Score** details. Product Score evaluates your product information and displays a score. It can be:

- Fair
- Good
- Excellent

- Best

Click **How do I improve my score?** to improve your product score.

8. After providing the product listing details, click **Save** before moving to the next section.

## 6.1. OVERVIEW

This tab consists of a series of tasks that you must complete to publish your product:

- [Section 6.1.1, "Complete product listing details"](#)
- [Section 6.1.2, "Complete company profile information"](#)
- [Section 6.1.3, "Certify your product"](#)
- [Section 6.1.4, "Add at least one product component"](#)
- [Section 6.1.5, "Certify components for your listing"](#)

### 6.1.1. Complete product listing details

1. To complete your product listing details, click **Start**.  
The **Product Information** tab opens.
2. Enter all the essential product details and click **Save**.

### 6.1.2. Complete company profile information

1. To complete your company profile information, click **Start**. After entering all the details, click **Submit**.
2. To modify the existing details, click **Review**. The **Account Details** page opens.
3. Review and modify the Company profile information and click **Submit**.

### 6.1.3. Certify your product

1. To certify your product, click **Certify product**. Read the **Publication and testing guidelines**.
  - a. Click **Add Component** and then go to [Section 6.1.4, "Add at least one product component"](#).

### 6.1.4. Add at least one product component

1. Click **Start**. You are redirected to the **Components** tab.  
To add a new or existing product component, click **Add component**.
2. For adding a new component,
  - a. In the **Component Name** text box, enter the component name.
  - b. For **What kind of standalone component are you creating?** select the component that you wish to certify. For example, for certifying a non containerized component, select **Non-containerized Application**.

- c. For **Red Hat Enterprise Linux Version** select the major RHEL version for which you are certifying your component.

**NOTE**

You can't modify the version after creating the product listing.

- d. Click **Create new component**
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

### 6.1.5. Certify components for your listing

1. To certify the components for your listing, click **Start**. If you have existing product components, you can view the list of **Attached Components** and their details:
  - a. Name
  - b. Certification
  - c. Security
  - d. Type
  - e. Created
  - f. Click more options to archive or remove the components
2. Select the components for certification.

After completing all the above tasks you will see a green tick mark corresponding to all the options.

The Overview tab also provides the following information:

1. **Product contacts** - Provides Product marketing and Technical contact information.
  - a. Click **Add contacts to product** to provide the contact information
  - b. Click **Edit** to update the information.
2. **Components in product** - Provides the list of the components attached to the product along with their last updated information.
  - a. Click **Add components to product** to add new or existing components to your product.
  - b. Click **Edit components** to update the existing component information.

After publishing the product listing, you can view your **Product Readiness Score** and **Ways to raise your score** on the **Overview** tab.

## 6.2. PRODUCT INFORMATION

Through this tab you can provide all the essential information about your product. The product details are published along with your product on the Red Hat Ecosystem catalog.

### General tab:

Provide 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. Click **Save**.

### Features & Benefits tab:

Provide important features of your product.

1. Optional: Enter the **Title** and **Description**.
2. Optional: To add additional features for your product, click + **Add new feature**.
3. Click **Save**.

### Quick start & Config tab:

Add links to any quick start guide or configuration document to help customers deploy and start using your product.

1. Optional: Enter **Quick start & configuration instructions**
2. Click **Save**.
3. Select **Hide default instructions** check box, if you don't want to display them.

### Linked resources tab:

Add links to supporting documentation to help our customers use your product. The information is mapped to and is displayed in the Documentation section on the product's catalog page.



#### NOTE

It is mandatory to add a minimum of three resources. Red Hat encourages you to add more resources, if available.

1. Select the **Type** drop-down menu, and enter the **Title** and **Description** of the resource.
2. Enter the **Resource URL**.
3. Optional: To add additional resources for your product, click + **Add new Resource**.
4. Click **Save**.

### FAQs tab:

Add frequently asked questions and answers of the product's purpose, operation, installation, or other attribute details. You can include common customer queries about your product and services.

1. Enter **Question** and **Answer**.
2. Optional: To add additional FAQs for your product, click **+ Add new FAQ**.
3. Click **Save**.

#### 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**.

#### Contacts tab:

Provide contact information of your marketing and technical team.

1. Enter the **Marketing contact email address** and **Technical contact email address**.
2. Optional: To add additional contacts, click **+ Add another**.
3. Click **Save**.

#### Legal tab:

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:

Use this tab to improve the 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.
3. Click **Save**.
4. Optional: To add additional key-value pair, click **+ Add new key-value pair**.



#### NOTE

Add at least one Search alias for your product. Red Hat encourages you to add more aliases, if available.

## 6.3. COMPONENTS

Use this tab to add components to your product listing. Through this tab you can also view a list of attached components linked to your Product Listing.

Alternatively, to attach a component to the Product Listing, you can complete the **Add at least one product component** option available on the **Overview** tab of a Container, Operator, or Helm Chart product listing.

1. To add a new or existing product component, click **Add component**.
2. For adding a new component, in the **Component Name** text box, enter the component name.
  - a. For **What kind of standalone component are you creating?** select the component that you wish to certify. For example, for certifying a non containerized component, select **Non-containerized Application**.
  - b. For **Red Hat Enterprise Linux Version** select the RHEL version on which you are certifying your non-containerized component.



#### NOTE

You cannot modify the RHEL version after creating the product listing.

- c. Click **Next**.
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.



#### NOTE

You can add the same component to multiple products listings. All attached components must be published before the product listing can be published.

After attaching components, you can view the list of **Attached Components** and their details:

- i. Name
- ii. Certification
- iii. Security
- iv. Type
- v. Created
- vi. Click more options to archive or remove the attached components

Alternatively, to search for specific components, type the component's name in the **Search by component Name** text box.

## 6.4. SUPPORT

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

You can also contact the Red Hat Partner Acceleration Desk for any technical questions you may have regarding the Certification. Technical help requests will be redirected to the Certification Operations team.

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.

1. To request support, click Open a support case. See [PAD - How to open & manage PAD cases](#), to open a PAD ticket.
2. To view the list of existing support cases, click [View support cases](#).

## 6.5. REMOVING A PRODUCT

After creating a product listing if you wish to remove it, go to the **Overview** tab and click **Delete**.

A published product must first be unpublished before it can be deleted. Red Hat retains information related to deleted products even after you delete the product.

# CHAPTER 7. ADDING CERTIFICATION COMPONENTS

After creating the new product listing, add the certification components for the newly created product listing. You can configure the following options for the newly added components:



## NOTE

The component configurations differ for different product categories.

- [Section 7.1, "Certification"](#)
- [Section 7.2, "Component Details"](#)
- [Section 7.3, "Contact Information"](#)
- [Section 7.4, "Associated products"](#)

To configure the component options, go to the **Components** tab and click on any of the existing components.

## 7.1. CERTIFICATION

### Verify the functionality of your product on Red Hat Enterprise Linux

Verify the functionality of your product on Red Hat Enterprise Linux by using the Certification tab. You can perform the following functions:

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 . Interact with the certification team, if required.

To verify the functionality of your product, perform the following steps:

1. If you are a new partner, click **Request a partner subscription**. When your request is approved, you get active subscriptions added to your account.
2. When you have active partner subscriptions, then click **Start certification**.
3. Click **Go to Red Hat certification tool**

A new certification case gets created on the [Red Hat Certification portal](#), and you are redirected to the appropriate component portal page.

The certification team will contact you to start the certification testing process and will follow up with you in case of a problem. After successful verification, a green check mark is displayed with the validate complete message.

To review the product details, click **Review**.

## 7.2. COMPONENT DETAILS

Enter the required component details in the following fields:

1. **Component name** - Enter the component name. This name is not published and is only for internal use.

2. **Red Hat Enterprise Linux (RHEL) Version-** Specifies the RHEL version on which you wish to certify your non-containerized product component.

**NOTE**

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

## 7.3. CONTACT INFORMATION

**NOTE**

Providing information for this tab is optional.

In the **Contact Information** tab, enter the primary technical contact details of your product component.

1. Optional: In the **Technical contact email address** field, enter the email address of the image maintainer.
2. Optional: To add additional contacts for your component, click **+ Add new contact**.
3. Click **Save**.

## 7.4. ASSOCIATED PRODUCTS

The Associated Product tab provides the list of products that are associated with your product component along with the following information:

- Product Name
- Type - Traditional application
- Visibility - Published or Not Published
- Last Activity - number of days before you ran the test

To add products to your component, perform the following:

- If you want to find a product by its name, enter the product name in the **Search by name** text box and click the search icon.
- If you are not sure of the product name, click **Find a product**. From the **Add product** dialog, select the required product from the Available products list box and click the forward arrow. The selected product is added to the Chosen products list box. Click **Update attached products**. Added products are listed in the Associated product list.

**NOTE**

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

# CHAPTER 8. 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.

## 8.1. 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 later 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

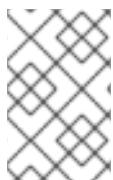
1. Configure the *Red Hat Certification* repository:
  - a. Use your RHN credentials to register your system using Red Hat Subscription Management:
 

```
$ 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:
 

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

Replace VERSION with the RHEL version number on the SUT

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

To find out the system architecture, run

```
$ uname -m
```

Replace HOSTTYPE with the system architecture.

```
$ subscription-manager repos --enable=cert-1-for-rhel-<VERSION>-x86_64-rpms
```

f. Install the software test suite package:



**NOTE**

This step is applicable only for RHEL version 9 or later.

```
$ dnf install redhat-certification-software
```

# CHAPTER 9. 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 10. RUNNING CERTIFICATION TESTS BY USING CLI AND DOWNLOADING THE RESULTS FILE

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

## 10.1. USING THE TEST PLAN TO PREPARE THE HOST UNDER TEST FOR TESTING

Running the provision command performs a number of operations, such as setting up passwordless SSH communication with the test server, installing the required packages on your system based on the certification type, and 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 hardware or software packages will be installed if the test plan is designed for certifying a hardware or a software product.

### Prerequisites

- You have the hostname or the IP address of the test server.

### Procedure

1. Run the provision command in either way. The test plan will automatically get downloaded to your system.

- If you have already downloaded the test plan:

```
# rhcert-provision <path_to_test_plan_document>
```

Replace <path\_to\_test\_plan\_document> with the test plan file saved on your system.

Follow the on-screen instructions.

- If you have not downloaded the test plan:

```
# rhcert-provision
```

Follow the on-screen instructions and enter your **Certification ID** when prompted.

2. When prompted, provide the hostname or the IP address of the test server to set up passwordless SSH. You are prompted only the first time you add a new system.

## 10.2. 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**.



#### NOTE

After a test reboot, **rhcrt** is running in the background to verify the image. Use **tail -f /var/log/rhcrt/RedHatCertDaemon.log** to see the current progress and status of the verification.

### 10.3. REVIEWING AND DOWNLOADING THE RESULTS FILE OF THE EXECUTED TEST PLAN

#### Procedure

1. Download the test results file:

```
# rhcert-save
```

2. Download the results file by using the **rhcrt-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 11. 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 12. 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

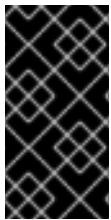
To recertify your application, it is mandatory to create a new certification request for recertification.

To recertify your application, submit a new certification request through the [Red Hat Certification tool](#) or create a new component in the [Red Hat Partner Connect](#). Run the certification tests on SUT and proceed with the regular certification workflow, like a new certification.

## CHAPTER 13. PUBLISHING THE CERTIFIED APPLICATION

After submitting your test results for certification through the [Red Hat certification portal](#), your application is scanned for vulnerabilities.

When the scanning is completed, the publish button will be enabled for your application on the [Product Listings](#) page. After providing all the necessary information, click the publish button. Your application will be available on the [Red Hat Ecosystem Catalog](#).



### IMPORTANT

Red Hat software certification or validation 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.

# APPENDIX A. RUNNING THE CERTIFICATION TESTS BY USING COCKPIT



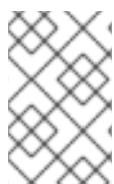
## NOTE

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 later 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.

```
# dnf install redhat-certification-cockpit
```

By default, Cockpit runs on port 9090.

#### Additional resources

For more information about 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 or later 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/](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/](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 **rhcrt-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/rhcrt/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/rhcrt/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/](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. CERTIFYING OR VALIDATING CONTAINERIZED APPLICATIONS

# CHAPTER 14. WORKING WITH CONTAINERS

## 14.1. 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.

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.

**Partner Validation** - Select this type of certification, if you want to validate your product using your own criteria and test suite on Red Hat platforms. This partner validation allows you to publish your software offerings on the Red Hat Ecosystem Catalog more quickly. However, validated workloads may not incorporate all of Red Hat integration requirements and best practices. We encourage you to continue your efforts toward Red Hat certification.

**Certified** - Select this type of certification, if you want your product to undergo thorough testing by using Red Hat's test suite, and benefit from collaborative support. Your products will meet your standards and Red Hat's criteria, including interoperability, lifecycle management, security, and support requirements.

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

It is critical for partners to understand the distinction between Partner Validation and Red Hat Certification. Find below a clear differentiation between Validation and Certification programs:

Feature	Partner Validation	Red Hat Certification
Testing Criteria	Determined by Partner	Determined by both Red Hat and Partner
Testing Tools	Provided by Partner	Provided by both Red Hat and Partner
Red Hat Review	Partner statements, supportive documentation and test results	Test results, partner statements, and documentation

<b>Scope</b>	Quicker market entry, initial presence on the Red Hat Ecosystem Catalog	Comprehensive compliance with Red Hat platform requirements, best practices such as interoperability, security, and support
--------------	-------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------

#### Additional resources

- [Introduction to Partner Validation](#)

## 14.2. CONTAINER CERTIFICATION WORKFLOW



### NOTE

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

#### Task Summary

The certification workflow includes the three primary stages-

1. [Section 14.2.1, "Certification on-boarding"](#)
2. [Section 14.2.2, "Certification testing for containerized applications"](#)
3. [Section 14.2.3, "Publishing the certified product listing on the Red Hat Ecosystem Catalog"](#)

### 14.2.1. Certification on-boarding

Perform the steps outlined for certification onboarding:

1. Join the [Red Hat Connect](#) for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Create your product listing by selecting your desired product category. You can select from the available product categories:
  - a. Containerized Application
  - b. Standalone Application
  - c. OpenStack Infrastructure
4. Complete your company profile.
5. Add components to the product listing.
6. Certify components for your product listing.

#### Additional resources

For detailed instructions about creating your first product listing, see [Creating a product](#).

### 14.2.2. Certification testing for containerized applications

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](#).

Additionally, if you are using a public registry—such as **DockerHub**, **GitLab**, **GitHub**, or similar that enforces rate limits on unauthenticated or anonymous access, provide a Docker configuration file with valid authentication credentials for the specified registry. Without this configuration, Red Hat certification systems can exceed the allowed request limits, preventing it from certifying or serving your container.

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](#).

### 14.2.3. Publishing the certified product listing on the Red Hat Ecosystem Catalog

The **Certified** container must be added 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.

#### Additional resources

- For more information about containers, see:

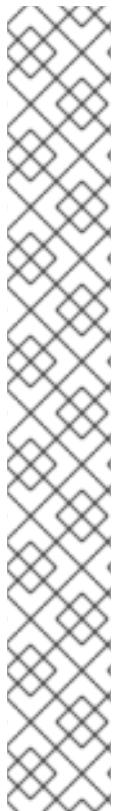
- [Choosing the right container image](#)
- [Everything you need to know about Red Hat Universal Base Image](#)

## 14.3. 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](#).

Additionally, if you are using a public registry—such as **DockerHub**, **GitLab**, **GitHub**, or similar that enforces rate limits on unauthenticated or anonymous access, provide a Docker configuration file with valid authentication credentials for the specified registry. Without this configuration, Red Hat certification systems can exceed the allowed request limits, preventing it from certifying or serving your container.

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](#).

### 14.3.1. 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 15. CREATE A PRODUCT

The product listing provides marketing and technical information, showcasing your product's features and advantages to potential customers. It lays the foundation for adding all necessary components to your product for certification.

### 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 before certification.

You must construct your container images so that they meet the certification criteria and policy. For more details, see [image content requirements](#). You can also use [Red Hat base images](#) for building your container images. See [Red Hat Enterprise Linux Container Compatibility Matrix](#) before matching your container images with the container hosts.

### Procedure

Red Hat recommends completing all optional fields in the listing tabs for a comprehensive product listing. More information helps mutual customers make informed choices.

Red Hat encourages collaboration with your product manager, marketing representative, or other product experts when entering information for your product listing.

Fields marked with an asterisk (\*) are mandatory.

### Procedure

1. Log in to the [Red Hat Partner Connect Portal](#).
2. Go to the Certified technology portal tab and click [Visit the portal](#).
3. On the header bar, click **Product management**.
4. From the **Listing and certification** tab click **Manage products**.
5. From the **My Products** page, click **Create Product**.  
A **Create New Product** dialog opens.
6. Enter the **Product name**.
7. From the **What kind of product would you like to certify?** drop-down, select the required product category and click **Create product**. For example, select **Containerized Application** for creating a containerized product listing.

A new page with your Product name opens. It comprises the following tabs:

- [Section 6.1, "Overview"](#)
- [Section 6.2, "Product Information"](#)
- [Section 6.3, "Components"](#)
- [Section 6.4, "Support"](#)

Along with the following tabs, the page header provides the **Product Score** details. Product Score evaluates your product information and displays a score. It can be:

- Fair
  - Good
  - Excellent
  - Best
8. Click **How do I improve my score?** to improve your product score.
  9. After providing the product listing details, click **Save** before moving to the next section.

## 15.1. OVERVIEW

This tab consists of a series of tasks that you must complete to publish your product:

- [Section 15.1.1, "Complete product listing details for containerized applications"](#)
- [Section 15.1.2, "Complete company profile information for containerized applications"](#)
- [Section 15.1.3, "Accept legal agreements for containerized applications"](#)
- [Section 15.1.4, "Certify your containerized applications"](#)
- [Section 15.1.5, "Add at least one product component for containerized applications"](#)
- [Section 15.1.6, "Certify components for your listing for containerized applications"](#)

### 15.1.1. Complete product listing details for containerized applications

1. To complete your product listing details, click **Start**.  
The **Product Information** tab opens.
2. Enter all the essential product details and click **Save**.

### 15.1.2. Complete company profile information for containerized applications

1. To complete your company profile information, click **Start**. After entering all the details, click **Submit**.
2. To modify the existing details, click **Review**. The **Account Details** page opens.
3. Review and modify the Company profile information and click **Submit**.

### 15.1.3. Accept legal agreements for containerized applications

To publish your product image, agree to the terms regarding the distribution of partner container images.

1. To accept the legal agreements, click **Start**.
2. To preview or download the agreement, click **Review**.

The **Red Hat Partner Connect Container Appendix** document displays. Read the document to know the terms related to the distribution of container images.

#### 15.1.4. Certify your containerized applications

1. Click **Certify product** Read the [Publication and testing guidelines](#).
  - a. Click **Add Component** and then [Section 15.1.5, "Add at least one product component for containerized applications"](#).

#### 15.1.5. Add at least one product component for containerized applications

1. Click **Start**. You are redirected to the **Components** tab.  
To add a new or existing product component, click **Add component**.
2. For adding a new component,
  - a. In the **Component Name** text box, enter the component name.
  - b. For **What kind of standalone component are you creating?** select the component that you wish to certify. For example, for certifying your containers, based on your requirements select from the following options:
    - i. Container Image
    - ii. Containerized application for RHEL
    - iii. Containerized application for OpenStack
  - c. Click **Next**.
  - d. For **Red Hat Enterprise Linux Version** select the major RHEL version for which you are certifying your component.



#### NOTE

You can't modify the version after creating the product listing.

- e. Click **Create new component**
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

#### 15.1.6. Certify components for your listing for containerized applications

1. To certify the components for your listing, click **Start**. If you have existing product components, you can view the list of **Attached Components** and their details:
  - a. Name
  - b. Certification
  - c. Security

- d. Type
  - e. Created
  - f. Click more options to archive or remove the components
2. Select the components for certification.

After completing all the above tasks you will see a green tick mark corresponding to all the options.

The Overview tab also provides the following information:

1. **Product contacts** - Provides Product marketing and Technical contact information.
  - a. Click **Add contacts to product** to provide the contact information
  - b. Click **Edit** to update the information.
2. **Components in product** - Provides the list of the components attached to the product along with their last updated information.
  - a. Click **Add components to product** to add new or existing components to your product.
  - b. Click **Edit components** to update the existing component information.

After publishing the product listing, you can view your **Product Readiness Score** and **Ways to raise your score** on the **Overview** tab.

## 15.2. PRODUCT INFORMATION

Through this tab you can provide all the essential information about your product. The product details are published along with your product on the Red Hat Ecosystem catalog.

**General** tab:

Provide 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. Click **Save**.

**Features & Benefits** tab:

Provide important features of your product.

1. Optional: Enter the **Title** and **Description**.
2. Optional: To add additional features for your product, click + **Add new feature**.
3. Click **Save**.

**Quick start & Config** tab:

Add links to any quick start guide or configuration document to help customers deploy and start using your product.

1. Optional: Enter **Quick start & configuration instructions**.
2. Click **Save**.
3. Select **Hide default instructions** check box, if you don't want to display them.

#### **Linked resources** tab:

Add links to supporting documentation to help our customers use your product. The information is mapped to and is displayed in the Documentation section on the product's catalog page.



#### **NOTE**

It is mandatory to add a minimum of three resources. Red Hat encourages you to add more resources, if available.

1. Select the **Type** drop-down menu, and enter the **Title** and **Description** of the resource.
2. Enter the **Resource URL**.
3. Optional: To add additional resources for your product, click **+ Add new Resource**.
4. Click **Save**.

#### **FAQs** tab:

Add frequently asked questions and answers of the product's purpose, operation, installation, or other attribute details. You can include common customer queries about your product and services.

1. Enter **Question** and **Answer**.
2. Optional: To add additional FAQs for your product, click **+ Add new FAQ**.
3. Click **Save**.

#### **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**.

#### **Contacts** tab:

Provide contact information of your marketing and technical team.

1. Enter the **Marketing contact email address** and **Technical contact email address**.
2. Optional: To add additional contacts, click **+ Add another**.
3. Click **Save**.

**Legal tab:**

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:**

Use this tab to improve the 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.
3. Click **Save**.
4. Optional: To add additional key-value pair, click + **Add new key-value pair**.

**NOTE**

Add at least one Search alias for your product. Red Hat encourages you to add more aliases, if available.

## 15.3. COMPONENTS TAB FOR CONTAINERS

Use this tab to add components to your product listing. Additionally, you can view a list of attached components linked to your Product Listing.

Select from the following options:

- [Section 15.3.1, "Container images"](#)
- [Section 15.3.2, "Containerized application for RHEL"](#)
- [Section 15.3.3, "Containerized application for OpenStack"](#)

Alternatively, to attach a component to the Product Listing, you can complete the **Add at least one product component** option available on the **Overview** tab of a Container, Operator, or Helm Chart product listing.

### 15.3.1. Container images

1. To add a new or existing product component, click **Add component**.
2. For adding a new component perform the steps,
  - a. In the **Component Name** text box, enter the component name.
  - b. For **What kind of standalone component are you creating?** select the component that you wish to certify. For example, for certifying your containers, select **Container Image**.
  - c. Click **Next**.

- d. On the **Create and Add** component page, select the preferred OS Content-Type and Distribution Method for the component:
  - i. For **What base image does your container use?**, select the type of image that you want to use for your component:
    - A. **Red Hat Universal Base Image** - You can distribute UBI-based container images through the Red Hat Container registry or any other third-party registry.
    - B. **Red Hat Enterprise Linux** - You can distribute RHEL-based container images through the Red Hat Container registry only.
  - ii. For **Select your preferred 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.
    - A. **Red Hat Container Registry** - Select this option if you want Red Hat to distribute your containers through Red Hat's container registry. If you select this option, images with this distribution method are hosted on your container registry, but are distributed to customers through a Red Hat registry proxy address. Customers will have access to your containers without adding registries to their configuration, but you will not have visibility regarding customer-specific download metrics or other usage data from the proxy.



#### NOTE

If you are a grandfathered partner, who chose the option **Red Hat Container Registry**, you will see an additional checkbox titled **I need Red Hat to host my registry**. This checkbox triggers the creation of a Red Hat managed quay.io registry for use with this component.

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



#### NOTE

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

- e. Click **Add Component**.
- 3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

### 15.3.2. Containerized application for RHEL

1. To add a new or existing product component, click **Add component**.
2. For adding a new component perform the steps,
  - a. In the **Component Name** text box, enter the component name.
  - b. For **What kind of standalone component are you creating?** select the component that you wish to certify. For example, select Containerized application for RHEL.
  - c. Click **Next**.
  - d. On the **Create and Add component** page, select the preferred RHEL version and Distribution Method for the component:
    - i. For **What major version of RHEL will you be certifying your image for?** select the preferred RHEL version.



#### NOTE

You cannot modify the version after creating the product component.

- ii. For **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.
  - A. **Red Hat Container Registry**- Select this option, if you want Red Hat to distribute your containers through Red Hat's container registry. Select the **I need Red Hat to host my registry** check box. When you select this option, images with this distribution method are hosted on your container registry but are distributed to customers through a Red Hat registry proxy address. 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.



#### NOTE

If you are a grandfathered partner, who chose the option **Red Hat Container Registry**, you will see an additional checkbox titled **I need Red Hat to host my registry**. This checkbox triggers the creation of a Red Hat managed quay.io registry for use with this component.

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

**NOTE**

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

- e. Click **Add Component**.
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

### 15.3.3. Containerized application for OpenStack

1. To add a new or existing product component, click **Add component**.
2. For adding a new component, in the **Component Name** text box, enter the component name.
  - a. For **What kind of standalone component are you creating?** select the component that you wish to certify. For example, for certifying a containerized application for Red Hat OpenStack platform, select **Containerized application for OpenStack**
  - b. For **What major version of OpenStack will you be certifying your image for?** Version 17 is enabled by default. You cannot modify this field.
  - c. Click **Create new Component**
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

**NOTE**

You can add one component to multiple products listings. All attached components must be published before the product listing can be published.

After attaching components, you can view the list of **Attached Components** and their details:

- i. Name
- ii. Certification
- iii. Security

- iv. Type
- v. Created
- vi. Click more options to archive or remove the attached components

Alternatively, to search for specific components, type the component's name in the **Search by component Name** text box.

## 15.4. SUPPORT

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

You can also contact the Red Hat Partner Acceleration Desk for any technical questions you may have regarding the Certification. Technical help requests will be redirected to the Certification Operations team.

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.

1. To request support, click Open a support case. See [PAD - How to open & manage PAD cases](#), to open a PAD ticket.
2. To view the list of existing support cases, click [View support cases](#).

## 15.5. REMOVING A PRODUCT

After creating a product listing if you wish to remove it, go to the **Overview** tab and click **Delete**.

A published product must first be unpublished before it can be deleted. Red Hat retains information related to deleted products even after you delete the product.

# CHAPTER 16. ADDING CERTIFICATION COMPONENTS

After creating the new product listing, add the certification components for the newly created product listing.

You can configure the following options for the newly added components:



## NOTE

The component configurations differ for different product categories.

- [Section 16.1, "Images"](#)
- [Section 16.2, "Certification for containers"](#)
- [Section 16.3, "Security"](#)
- [Section 16.4, "Repository information"](#)
- [Section 16.5, "Component details"](#)
- [Section 7.3, "Contact Information"](#)
- [Section 16.7, "Associated products for containers"](#)

To configure the options, go to the **Components** tab and click on any of the existing components.

## 16.1. IMAGES

The Images tab provides the test results for the container images that you submit by using the preflight tool. You have to configure preflight and push your container images to view the test results.

- To push your container images, click **Set up Preflight**.
- For detailed instructions about certification testing, see [Running the certification test suite](#).

When your testing is complete you can see two categories of images:

- **Manifest Digests** - denotes container images that are available for multiple architectures.
- **Standalone Container Images** - denotes container images that are available only for a single architecture.

This page provides the following details of your container images:

- Specific image ID or the SHA ID
- Image Tag(s)
- Certification - Certified or Not certified, pass or fail status based on the checks performed. Click on it for more details.
- Architecture - specific architecture of your image, if applicable.
- Security - check for any vulnerabilities, if any.

- 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.
- Created - the day on which you submitted the certification.
- Click the Actions menu 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](#).
- Click **Publish**, to publish your certified container images.

## 16.2. CERTIFICATION FOR CONTAINERS

### 16.2.1. For Container images

The Certification tab provides detailed information about the Export control questionnaire, all the certification tests performed for the attached container images and ways to submit your container image for certification.

- Export Control Questionnaire

The [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 declines partner or defers decision or requests more information.

1. Click **Start questionnaire**, to enter all the legal information about your product.
2. Click **Review** to modify the existing details.



#### NOTE

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

- Certification tests

It provides the status of the [Manifest Digests Certification Tests](#) or [Standalone Certification Tests](#) performed for the attached container images, which includes the following details:

- Results - total number of tests run along with the result. Click on it for more details.
- Image - specific image ID or SHA ID
- Last activity - number of days before you ran the test

- Submit your container image for verification
  - Run the certification suite on your container image. See [Running the certification test suite](#).
  - Upload the test results. You can later see the test results on the **Images** tab.
  - Publish the container image certification on the Red Hat catalog. See [Publishing the certified container on Red Hat Ecosystem Catalog](#).



#### NOTE

This step certifies your container only. Use the **Certifications** tab to certify the functionality.

### 16.2.2. For Containerized applications on RHEL

- Export Control Questionnaire

The [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 declines partner or defers decision or requests more information.

1. Click **Start questionnaire**, to enter all the legal information about your product.
2. Click **Review** to modify the existing details.



#### NOTE

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

- Verify the functionality of your product on Red Hat Enterprise Linux  
Verify the functionality of your product on Red Hat Enterprise Linux by using the Certification tab. You can 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.
- Interact with the certification team, if required.

To verify the functionality of your product perform the following steps:

- If you are a new partner, click **Request a partner subscription**. When your request is approved, you get active subscriptions added to your account.
- When you have active partner subscriptions, then click **Start certification** and then click **Go to Red Hat certification tool**  
A new certification case gets created on the [Red Hat Certification portal](#), and you are redirected to the appropriate certification portal page.

The certification team will contact you to start the certification testing process, and will follow up with you in case of a problem. After successful verification, a green check mark is displayed with the validate complete message.

To review the product details, click **Review**.

- Submit your container image for verification
  - Run the certification suite on your container image. See [Running the certification test suite](#).
  - Upload the test results.  
You can later see the test results on the **Images** tab.
  - Publish the container image certification on the Red Hat catalog. See [Publishing the certified container on Red Hat Ecosystem Catalog](#).



#### NOTE

This step certifies your container only. Use the **Certifications** tab to certify the functionality.

### 16.2.3. For Containerized applications for OpenStack

The Certification tab provides detailed information about the Export control questionnaire and ways to submit your container image for certification.

- Export Control Questionnaire  
The [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 declines partner or defers decision or requests more information.
  1. Click **Start questionnaire**, to enter all the legal information about your product.
  2. Click **Review** to modify the existing details.



#### NOTE

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

- Submit your container image for verification
  - Run the certification suite on your container image. See [Running the certification test suite](#).
  - Upload the test results. You can later see the test results on the **Images** tab.
  - Publish the container image certification on the Red Hat catalog. See [Publishing the certified container on Red Hat Ecosystem Catalog](#).



### NOTE

This step certifies your container only. Use the **Certifications** tab to certify the functionality.

## 16.3. SECURITY

The security tab provides the health status of the attached product components. Red Hat uses a Health Index to identify security risks with your components that Red Hat provides through the [Red Hat Ecosystem Catalog](#).

The Health Index is a measure of the oldest and most severe security updates available for a container image. An image with a grade of 'A' is more up-to-date than one with a grade of 'F'. For more information, see [Container Health Index grades as used inside the Red Hat Container Catalog](#).

This tab provides the health index of your images which includes the following details:

- Image ID
- Health index

## 16.4. REPOSITORY INFORMATION

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

Enter the required details in the following fields:

Field name	Description
Container registry namespace	Registry name set when the container was created. This field becomes non-editable when the container gets published.
Outbound repository name	Repository name that you have selected or the name obtained from your private registry in which your image is hosted, for example, ubi-minimal.
Repository summary	Repository summary obtained from the container image.
Repository description	Repository description obtained from the container image.
Instructions for users to get your company's image on the Red Hat Container catalog	Provide specific instructions that you want users to follow when they get your container image. This field is applicable only for container images.

After configuring all the mandatory fields click **Save**.

**NOTE**

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

## 16.5. COMPONENT DETAILS

Configure the product component details by using this tab.

Enter the required details in the following fields:

Field name	Description
Image Type	<p>Select the respective image type for your product component.</p> <ul style="list-style-type: none"> <li>● <b>Operator image</b> - Select this type if you want to deploy an operator that manages other images.</li> <li>● <b>Standalone image</b> - Select this type if you want your image to be deployed either by your product or by users.</li> <li>● <b>Component image</b> - Select this type if you want your image to be deployed by your product and not by users.</li> </ul>
Application categories	<p>Select the respective application type of your software product.</p>
Host level access	<p>Select between the two options:</p> <ul style="list-style-type: none"> <li>● <b>Unprivileged</b> - If your container is isolated from the host. or</li> <li>● <b>Privileged</b> - If your container requires special host-level privileges.</li> </ul> <div data-bbox="806 1558 910 1783" style="background-color: #e0e0e0; width: 64px; height: 64px; margin-left: auto; margin-right: auto;"></div> <div data-bbox="985 1563 1083 1599" style="text-align: center;"><b>NOTE</b></div> <p>If your product's functionality requires root access, you must select the <b>privileged</b> option, before running the preflight tool. This setting is subject to Red Hat review.</p>

Field name	Description
Release Category	<p>Select between the two options:</p> <ul style="list-style-type: none"> <li>● <b>Generally Available</b> - When you select this option, the application is generally available and supported. or</li> <li>● <b>Beta</b> - When you select this option, the application is available as a pre-release candidate.</li> </ul>
Component name	Name of the component for internal purposes.
Auto-publish	When you enable this option, the container image gets automatically published on the Red Hat Container catalog, after passing all the certification tests.
Red Hat Enterprise Linux Version	<p>It denotes the version of RHEL on which you are certifying your containerized application.</p> <div style="display: flex; align-items: center;">  <b>NOTE</b>  <p>This field is non-editable and is applicable only for containerized applications on RHEL.</p> </div>
Red Hat OpenStack platform	<p>It denotes the version of the OpenStack platform on which you are certifying your containerized application.</p> <div style="display: flex; align-items: center;">  <b>NOTE</b>  <p>This field is non-editable and is applicable only for containerized applications on Red Hat OpenStack platform.</p> </div>

## 16.6. CONTACT INFORMATION



### NOTE

Providing information for this tab is optional.

In the **Contact Information** tab, enter the primary technical contact details of your product component.

1. Optional: In the **Technical contact email address** field, enter the email address of the image maintainer.
2. Optional: To add additional contacts for your component, click **+ Add new contact**.
3. Click **Save**.

## 16.7. ASSOCIATED PRODUCTS FOR CONTAINERS

The Associated Product tab provides the list of products that are associated with your product component along with the following information:

- Product Name
- Type
- Visibility - Published or Not Published
- Last Activity - number of days before you ran the test

To add products to your component, perform the following:

- If you want to find a product by its name, enter the product name in the **Search by name** text box and click the search icon.
- If you are not sure of the product name, click **Find a product**. From the **Add product** dialog, select the required product from the Available products list box and click the forward arrow. The selected product is added to the Chosen products list box. Click **Update attached products**. Added products are listed in the Associated product list.



### NOTE

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

# CHAPTER 17. RUNNING THE CERTIFICATION TEST SUITE

## 17.1. ALL CONTAINERIZED APPLICATIONS

To run the certification test suite for OpenShift and Red Hat Enterprise Linux (RHEL) containerized applications, you must complete all the steps in this section. For RHEL containerized applications, you must also complete the [Section 17.2, “Additional steps for RHEL containerized applications”](#) in order to run the certification 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>
```

The authentication file generated by using the **--authfile ./temp-authfile.json** option is required in the following steps. This authentication file is used by the **--docker-config** option when you submit the test results by using the Preflight tool.

- You have set up your container on the [Red Hat Partner Connect portal](#). The product listing 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 \
--docker-config=./temp-authfile.json \
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> \
--submit \
--pyxis-api-token=<api_token> \
--certification-component-id=<component_id> \
--docker-config=./temp-authfile.json
```

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 component UI by navigating to the *Images* tab in the [Red Hat Partner Connect portal](#).



#### NOTE

For RHEL containerized applications, you must also complete the [Section 17.2, "Additional steps for RHEL containerized applications"](#).

## 17.2. ADDITIONAL STEPS FOR RHEL CONTAINERIZED APPLICATIONS

To certify a RHEL containerized application, you have to complete all the previous steps in [Section 17.1, "All containerized applications"](#), then verify 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 verify your container.

#### Procedure

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

1. Install the preflight package:  
**# dnf install redhat-certification-preflight**
2. Run rhcert and follow the instructions:  
**# rhcert-run**
3. Review and save the test results:  
**# rhcert-save**

## CHAPTER 18. PUBLISHING THE CERTIFIED CONTAINER ON RED HAT ECOSYSTEM CATALOG

### 18.1. PUBLISHING CERTIFIED CONTAINERS

After you submit your test results from the preflight tool on your [Partner Connect portal](#), your container images are scanned for vulnerabilities. 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

Red Hat software certification or validation 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.

## PART III. CERTIFYING OR VALIDATING OPERATORS

# CHAPTER 19. WORKING WITH OPERATORS



## NOTE

Certify your operator image or necessary container image as a component 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.

## 19.1. 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 Operator Lifecycle Manager (OLM) on the OpenShift platform and is formatted properly, using Red Hat certified container images.

**Partner Validation** - Select this type of certification, if you want to validate your product using your own criteria and test suite on Red Hat platforms. This partner validation allows you to publish your software offerings on the Red Hat Ecosystem Catalog more quickly. However, validated workloads may not incorporate all of Red Hat integration requirements and best practices. We encourage you to continue your efforts toward Red Hat certification.

**Certified** - Select this type of certification, if you want your product to undergo thorough testing by using Red Hat's test suite, and benefit from collaborative support. Your products will meet your standards and Red Hat's criteria, including interoperability, lifecycle management, security, and support requirements.

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

It is critical for partners to understand the distinction between Partner Validation and Red Hat Certification. Find below a clear differentiation between Validation and Certification programs:

Feature	Partner Validation	Red Hat Certification
Testing Criteria	Determined by Partner	Determined by both Red Hat and Partner
Testing Tools	Provided by Partner	Provided by both Red Hat and Partner
Red Hat Review	Partner statements, supportive documentation and test results	Test results, partner statements, and documentation
Scope	Quicker market entry, initial presence on the Red Hat Ecosystem Catalog	Comprehensive compliance with Red Hat platform requirements, best practices such as interoperability, security, and support

## Additional resources

- [Introduction to Partner Validation](#)

## 19.2. 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 19.2.1, "Certification on-boarding for Operators"](#)
2. [Section 19.2.2, "Certification testing for Operators"](#)
3. [Section 19.2.3, "Publishing the certified Operator on the Red Hat Ecosystem Catalog"](#)

### 19.2.1. Certification on-boarding for Operators

Perform the steps outlined for certification onboarding:

1. Join the [Red Hat Connect](#) for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Create your product listing by selecting your desired product category. You can select from the available product categories:
  - a. Containerized Application
  - b. Standalone Application
  - c. OpenStack Infrastructure
4. Complete your company profile.
5. Add components to the product listing.
6. Certify components for your product listing.

#### Additional resources

- [Creating a product](#).

### 19.2.2. Certification testing for Operators

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.



### NOTE

It is possible that some operator releases seemingly disappear from the catalog, which happens when the graph gets automatically pruned, resulting in some operator versions being excluded from the update graph. Because of that, you will get blocked from releasing an operator bundle when it results in a channel with fewer or equal release versions than the one before.

In the case that you want to prune the graph intentionally, you can do so by skipping a test and restarting the pipeline using the following available commands in your pull request:

<b>/test skip &lt;test_case_name&gt;</b>	test_case_name test will be skipped. Note that only a subset of tests can be skipped.
<b>/pipeline restart certified-hosted-pipeline</b>	The hosted pipeline will re-trigger.

### Additional resources

- [Appendix B. Running the certification test suite locally](#) .

### 19.2.3. Publishing the certified Operator on the Red Hat Ecosystem Catalog

The **Certified** Operator must be added 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.

### Additional resources

- [Operators](#)
- [Operator Framework](#)
- [Operator Capability Levels](#)
- [Packaging Applications and Services with Kubernetes Operators](#)

# CHAPTER 20. CREATE A PRODUCT

The product listing provides marketing and technical information, showcasing your product's features and advantages to potential customers. It lays the foundation for adding all necessary components to your product for certification.

## 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 before certification.

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

## Procedure

Red Hat recommends completing all optional fields in the listing tabs for a comprehensive product listing. More information helps mutual customers make informed choices.

Red Hat encourages collaboration with your product manager, marketing representative, or other product experts when entering information for your product listing.

Fields marked with an asterisk (\*) are mandatory.

## Procedure

1. Log in to the [Red Hat Partner Connect Portal](#).
2. Go to the Certified technology portal tab and click **Visit the portal**.
3. On the header bar, click **Product management**.
4. From the **Listing and certification** tab click **Manage products**.
5. From the **My Products** page, click **Create Product**.  
A **Create New Product** dialog opens.
6. Enter the **Product name**.
7. From the **What kind of product would you like to certify?** drop-down, select the required product category and click **Create product**. For example, select **Containerized Application** for creating a containerized product listing.

A new page with your Product name opens. It comprises the following tabs:

- [Section 6.1, "Overview"](#)
- [Section 6.2, "Product Information"](#)
- [Section 6.3, "Components"](#)
- [Section 6.4, "Support"](#)

Along with the following tabs, the page header provides the **Product Score** details. Product Score evaluates your product information and displays a score. It can be:

- Fair

- Good
  - Excellent
  - Best
8. Click **How do I improve my score?** to improve your product score.
  9. After providing the product listing details, click **Save** before moving to the next section.

## 20.1. OVERVIEW

This tab consists of a series of tasks that you must complete to publish your product:

- [Section 20.1.1, "Complete product listing details for Operators"](#)
- [Section 20.1.2, "Complete company profile information for Operators"](#)
- [Section 20.1.3, "Accept legal agreements for Operators"](#)
- [Section 20.1.4, "Certify your Operators"](#)
- [Section 20.1.5, "Add at least one product component for Operators"](#)
- [Section 20.1.6, "Certify components for your listing for Operators"](#)

### 20.1.1. Complete product listing details for Operators

1. To complete your product listing details, click **Start**.  
The **Product Information** tab opens.
2. Enter all the essential product details and click **Save**.

### 20.1.2. Complete company profile information for Operators

1. To complete your company profile information, click **Start**. After entering all the details, click **Submit**.
2. To modify the existing details, click **Review**. The **Account Details** page opens.
3. Review and modify the Company profile information and click **Submit**.

### 20.1.3. Accept legal agreements for Operators

To publish your product image, agree to the terms regarding the distribution of partner container images.

1. To accept the legal agreements, click **Start**.
2. To preview or download the agreement, click **Review**.

The **Red Hat Partner Connect Container Appendix** document displays. Read the document to know the terms related to the distribution of container images.

### 20.1.4. Certify your Operators

1. Click **Certify product**. Read the [Publication and testing guidelines](#).
  - a. Click **Add Component** and then go to [Section 20.1.5, "Add at least one product component for Operators"](#).

## 20.1.5. Add at least one product component for Operators

1. Click **Start**. You are redirected to the **Components** tab.  
To add a new or existing product component, click **Add component**.
2. For adding a new component,
  - a. In the **Component Name** text box, enter the component name.
  - b. For **What kind of OpenShift component are you creating?** select the component that you wish to certify. For example, for certifying your operators, select **Operator Bundle**.
  - c. Click **Next**.
  - d. **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. ...Select the required operator:
      - A. Container Network Interface (CNI)
      - B. Cloud Storage Interface (CSI)
  - e. **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 operator component 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](#) and published to the certified operator index embedded in the OperatorHub of OpenShift.
  - f. Click **Add component**.
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

## 20.1.6. Certify components for your listing for Operators

1. To certify the components for your listing, click **Start**. If you have existing product components, you can view the list of **Attached Components** and their details:

- a. Name
  - b. Certification
  - c. Security
  - d. Type
  - e. Created
  - f. Click more options to archive or remove the components
2. Select the components for certification.

After completing all the above tasks you will see a green tick mark corresponding to all the options.

The Overview tab also provides the following information:

1. **Product contacts** - Provides Product marketing and Technical contact information.
  - a. Click **Add contacts to product** to provide the contact information
  - b. Click **Edit** to update the information.
2. **Components in product** - Provides the list of the components attached to the product along with their last updated information.
  - a. Click **Add components to product** to add new or existing components to your product.
  - b. Click **Edit components** to update the existing component information.

After publishing the product listing, you can view your **Product Readiness Score** and **Ways to raise your score** on the **Overview** tab.

## 20.2. PRODUCT INFORMATION

Through this tab you can provide all the essential information about your product. The product details are published along with your product on the Red Hat Ecosystem catalog.

**General** tab:

Provide 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. Click **Save**.

**Features & Benefits** tab:

Provide important features of your product.

1. Optional: Enter the **Title** and **Description**.
2. Optional: To add additional features for your product, click **+ Add new feature**.

3. Click **Save**.

#### Quick start & Config tab:

Add links to any quick start guide or configuration document to help customers deploy and start using your product.

1. Optional: Enter **Quick start & configuration instructions**
2. Click **Save**.
3. Select **Hide default instructions** check box, if you don't want to display them.

#### Linked resources tab:

Add links to supporting documentation to help our customers use your product. The information is mapped to and is displayed in the Documentation section on the product's catalog page.



#### NOTE

It is mandatory to add a minimum of three resources. Red Hat encourages you to add more resources, if available.

1. Select the **Type** drop-down menu, and enter the **Title** and **Description** of the resource.
2. Enter the **Resource URL**.
3. Optional: To add additional resources for your product, click **+ Add new Resource**.
4. Click **Save**.

#### FAQs tab:

Add frequently asked questions and answers of the product's purpose, operation, installation, or other attribute details. You can include common customer queries about your product and services.

1. Enter **Question** and **Answer**.
2. Optional: To add additional FAQs for your product, click **+ Add new FAQ**.
3. Click **Save**.

#### 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**.

#### Contacts tab:

Provide contact information of your marketing and technical team.

1. Enter the **Marketing contact email address** and **Technical contact email address**.

2. Optional: To add additional contacts, click + **Add another**.

3. Click **Save**.

#### Legal tab:

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:

Use this tab to improve the 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.

3. Click **Save**.

4. Optional: To add additional key-value pair, click + **Add new key-value pair**.



#### NOTE

Add at least one Search alias for your product. Red Hat encourages you to add more aliases, if available.

### 20.3. COMPONENTS

Use this tab to add components to your product listing. Through this tab you can also view a list of attached components linked to your Product Listing.

Alternatively, to attach a component to the Product Listing, you can complete the **Add at least one product component** option available in the **Overview** tab of a Container, Operator, or Helm Chart product listing.

1. To add a new or existing product component, click **Add component**.

2. For adding a new component,

a. In the **Component Name** text box, enter the component name.

b. For **What kind of OpenShift component are you creating?** select the component that you wish to certify. For example, for certifying your operators, select Operator Bundle.

c. Click **Next**.

d. **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)
- e. **Publication Options** - Select one of the following options for publishing your operator:
- A. **Web catalog only ([catalog.redhat.com](#))** - The operator is published to the [Red Hat Container Catalog](#) but is not visible on Red Hat OpenShift OperatorHub. This is the default option when you create a new operator bundle component.
  - B. **OpenShift In-product Catalog (Certified)** - The operator is listed on the [Red Hat Container Catalog](#) and published to the certified operator index embedded in the OperatorHub of OpenShift. This option gives customers the availability of installing your operator directly from OperatorHub in the OpenShift UI.
- f. Click **Add Component**.
3. For adding an existing component, from the Add Component dialog, select **Existing Component**.
- a. From the **Available components** list, search and select the components that you wish to add and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.



#### NOTE

You can add the same component to multiple products listings. All attached components must be published before the product listing can be published.

After attaching components, you can view the list of **Attached Components** and their details:

- i. Name
- ii. Certification
- iii. Security
- iv. Type
- v. Created
- vi. Click more options to archive or remove the attached components

Alternatively, to search for specific components, type the component's name in the **Search by component Name** text box.

## 20.4. SUPPORT

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

You can also contact the Red Hat Partner Acceleration Desk for any technical questions you may have regarding the Certification. Technical help requests will be redirected to the Certification Operations team.

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.

1. To request support, click Open a support case. See [PAD - How to open & manage PAD cases](#), to open a PAD ticket.
2. To view the list of existing support cases, click [View support cases](#).

## 20.5. REMOVING A PRODUCT

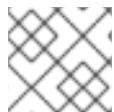
After creating a product listing if you wish to remove it, go to the **Overview** tab and click **Delete**.

A published product must first be unpublished before it can be deleted. Red Hat retains information related to deleted products even after you delete the product.

# CHAPTER 21. ADDING CERTIFICATION COMPONENTS

After creating the new product listing, add the certification components for the newly created product listing.

You can configure the following options for the newly added components:



## NOTE

The component configurations differ for different product categories.

- [Section 21.1, “Certification for Operators”](#)
- [Section 21.2, “Optional Qualifications for Operators”](#)
- [Section 21.3, “Repository Information for Operators”](#)
- [Section 21.4, “Component details for Operators”](#)
- [Section 21.5, “Contact Information for Operators”](#)
- [Section 21.6, “Associated products for Operators”](#)
- [Section 21.7, “Update Graph”](#)

To configure the options, go to the **Components** tab and click on any of the existing components.

## 21.1. CERTIFICATION FOR OPERATORS

- Verify the functionality of your CNI or CSI on Red Hat OpenShift



## NOTE

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 verify the functionality of your specialized CNI or CSI operator:

1. Click **Go to Red Hat certification tool** A new certification case gets created on the Red Hat Certification portal after which you are redirected to the appropriate 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 verify your specialized CNI or CSI operator. Upon successful validation, your operator will get approved and published.
- [CNI](#) and [CSI](#) workflow.
  - Operator Certification

To run the Operator certification suite, go to Testing Options. It displays two tabs to determine how to test and certify your operator.

- 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 [Appendix B. Running the certification test suite locally](#).
- 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 the CI Pipeline, for testing your Operator. This method allows you to incorporate the tests into your CI/CD workflows and development processes, therefore ensuring the proper functioning of your product on the OpenShift platform and streamlining 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 for several use cases, such as when working on an urgent deadline, or when enough resources and time are not available to learn and use the tooling.

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

- **Most recent test run tab** provides the latest test results, if any.

The certification table provides the following information:

- Operator version
- Pull request
- Tested on
- Test result - Pass or Fail
- Created
- Click **View all tests** for detailed information about all the tests. It has two tabs:
  - **Test Results** - Displays a summary of all the certification tests along with their results.
  - **Test Artifacts** - Displays log files.

## 21.2. OPTIONAL QUALIFICATIONS FOR OPERATORS



### NOTE

This tab is applicable only for Operator and Helm chart certifications.

The **Optional qualifications** tab provides the option to verify if your product follows Red Hat's recommended guidelines and best practices for deploying workload on Red Hat OpenShift. When you select this tab, a functional certification is created where you will submit testing results for Red Hat's review. After successful verification your workload product gets listed as Certified with the **Meets Best Practices** badge on the Red Hat Ecosystem catalog.

### Additional resources

For more information, see [Best practices](#)

## 21.3. REPOSITORY INFORMATION FOR OPERATORS

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

Enter the required details in the following fields:

Field name	Description
Container registry namespace	Registry name set when the container was created. This field becomes non-editable when the container gets published.
Outbound repository name	Repository name that you have selected or the name obtained from your private registry in which your image is hosted, for example, <i>ubi-minimal</i> .
Authorized GitHub user accounts	It denotes the GitHub users who are allowed to submit operators for certification on behalf of your company.
Repository summary	Repository summary obtained from the container image.
Repository description	Repository description obtained from the container image.

After configuring all the mandatory fields click **Save**.



### NOTE

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

## 21.4. COMPONENT DETAILS FOR OPERATORS

Configure the product component details by using this tab.

Enter the required details in the following fields:

Field name	Description
Image Type	Operator bundle is selected by default.
Application categories	Select the respective application type of your software product.
Component name	Name of the component for internal purposes.

After configuring all the mandatory fields click **Save**.

## 21.5. CONTACT INFORMATION FOR OPERATORS



### NOTE

Providing information for this tab is optional.

In the **Contact Information** tab, enter the primary technical contact details of your product component.

1. Optional: In the **Technical contact email address** field, enter the email address of the image maintainer.
2. Optional: To add additional contacts for your component, click **+ Add new contact**.
3. Click **Save**.

## 21.6. ASSOCIATED PRODUCTS FOR OPERATORS

The Associated Product tab provides the list of products that are associated with your product component along with the following information:

- Product Name
- Type
- Visibility - Published or Not Published
- Last Activity - number of days before you ran the test

To add products to your component, perform the following:

- If you want to find a product by its name, enter the product name in the **Search by name** text box and click the search icon.
- If you are not sure of the product name, click **Find a product**. From the **Add product** dialog, select the required product from the Available products list box and click the forward arrow. The selected product is added to the Chosen products list box. Click **Update attached products**. Added products are listed in the Associated product list.

**NOTE**

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

## 21.7. UPDATE GRAPH

Select the OpenShift product version and Channel details for your component through this tab.

- Select the required version from the **OpenShift Version** list box.
- Select the required channel from the **Channel** list box.

The Update graph table provides the following information:

- Version
- Update Paths
- Other Available Channels

See [Operator update documentation](#) tile below the header, for more information about the upgrades.

**NOTE**

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

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

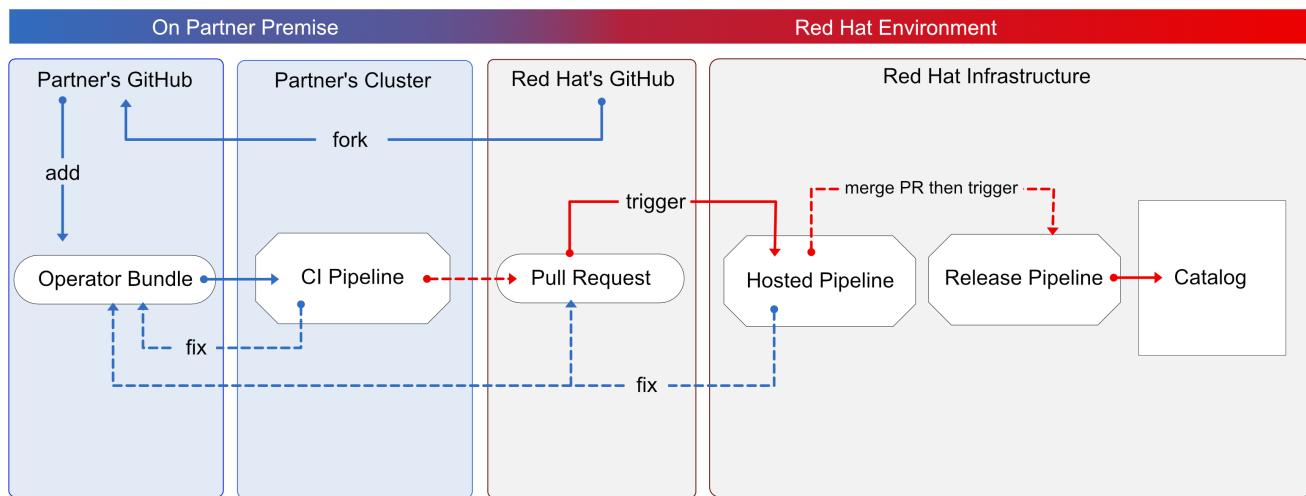
If you want to certify your operator with the Red Hat Hosted Pipeline you have to create a pull request for the Red Hat certification repository.

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.

Red Hat recommends running the certification test suite locally to certify your operators. For more details, see [Appendix B. Running the certification test suite locally](#).

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. 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 *Product listing* available on the [Red Hat Partner Connect](#) website.
- On the [Red Hat Partner Connect](#) website, go to **Components** tab.
- In the **Authorized GitHub user accounts** field, enter your GitHub username to the list of authorized GitHub users.

### 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:

Catalog	Upstream Repository
Certified Catalog	<a href="https://github.com/redhat-openshift-ecosystem/certified-operators">https://github.com/redhat-openshift-ecosystem/certified-operators</a>

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](#).

## 22.2. ADDING YOUR OPERATOR BUNDLE

Add your operator bundle by using any of the two methods, depending on the workflow:

- **File-Based Catalog (FBC) workflow**

If you are using the FBC process, continue with [File-based Catalog \(FBC\)](#).



### NOTE

Use File-Based Catalog workflow for all new operator certifications and to convert all the existing certified operators to a more scalable, template-driven format.

- **Classic (non-FBC) workflow** If you are using the classic, directory-based process, proceed with:
  - [Section 22.2.1, "If you have certified this operator before"](#)
  - [Section 22.2.2, "If you are newly certifying this operator"](#)

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

Find your operator folder in the operators directory. Place the contents of your operator bundle in this directory.



### NOTE

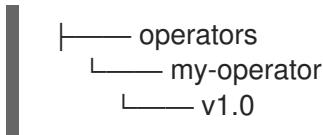
Make sure your package name is consistent with the existing folder name for your operator.

## 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**.

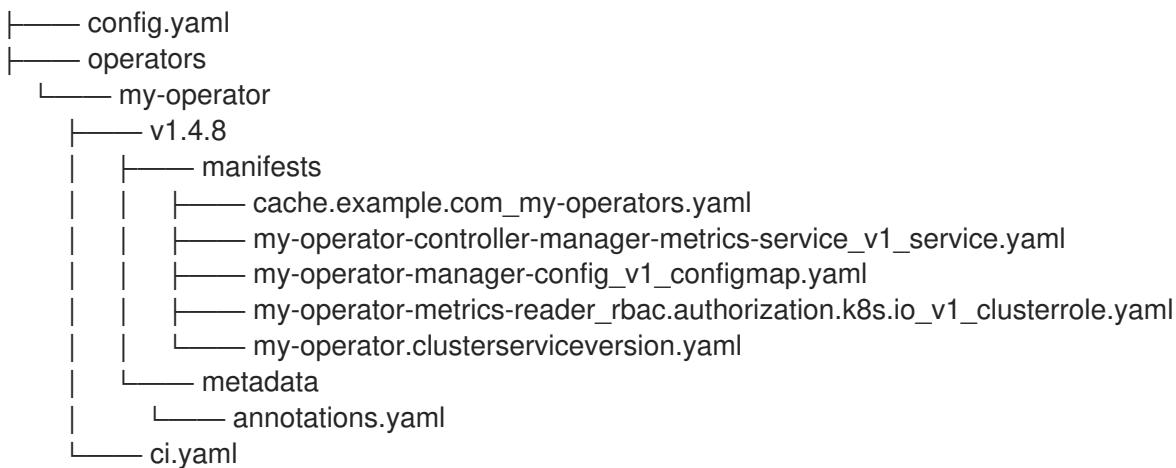
- In this operators directory, create a new subdirectory with the name of your operator, for example, <**my-operator**> and create a version directory for example, <**V1.0**> and place your bundle. The certification process preinstalls these directories for operators that were previously certified.



- 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.



Configuration file	Description
config.yaml	In this file include the organization of your operator. It can be <b>certified-operators</b> . For example, <b>organization: certified-operators</b>
ci.yaml	In this file include your Red Hat Technology Partner Component PID for this operator.  For example, <b>cert_project_id: &lt;your component pid&gt;</b> . This file stores all the necessary metadata for a successful certification process.

Configuration file	Description
annotations.yaml	<p>In this file include an annotation of OpenShift versions, which refers to the range of OpenShift versions . For example, <b>v4.8-v4.10</b> means versions 4.8 through 4.10. Add this to any existing content.</p> <p>For example, <b># OpenShift annotations</b>  <b>com.redhat.openshift.versions: v4.8-v4.10.</b></p> <p>The <b>com.redhat.openshift.versions</b> 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.</p> <p>Note that the letter 'v' must be used before the version, and spaces are not allowed. The syntax is as follows:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• A single version preceded by '=' indicates that the operator is supported only on that specific version of OpenShift. For example, using <b>=v4.8</b> will add the operator to the certified catalog for OpenShift 4.8, but not for later OpenShift releases.</li> <li>• A range can be used to indicate support only for OpenShift versions within that range. For example, using <b>v4.8-v4.10</b> will add the operator to the certified catalog for OpenShift 4.8 through 4.10, but not for OpenShift 4.11 or 4.12.</li> </ul>

## Additional resources

- For more details, see [Managing OpenShift Versions](#).
- For more information about building an operator bundle, see [Working with bundle images](#).
- For an example of an operator bundle, see [here](#).

## 22.3. FILE-BASED CATALOG

The File-Based Catalog (FBC) provides a plain text, declarative configuration format for packaging and managing operator upgrade paths on OpenShift clusters.

A key limitation of the older format is that operators could not modify their upgrade graphs after releasing a bundle. FBC addresses this by allowing changes to upgrade paths without requiring a new bundle release, offering more flexibility in certification and post-release updates.

### 22.3.1. Onboarding

You should convert your operator to the FBC format before submitting it through the certification workflow. The onboarding process uses automation to simplify the conversion from a bundle-based format to FBC.

#### Prerequisites

Before starting the onboarding procedure, ensure the following:

- You are in the correct operator directory:

```
cd <operator-repo>/operators/<operator-name>
```

- The following dependencies are installed:
  - **podman**
  - **make**
- You are authenticated to the required registries used by the Operator Lifecycle Manager (OLM). Use **podman login** and verify that registry credentials are stored in either:
  - **\$(XDG\_RUNTIME\_DIR)/containers/auth.json** or
  - **~/.docker/config.json**

#### Procedure

1. Download the **Makefile** to automate the onboarding process:

```
wget https://raw.githubusercontent.com/redhat-openshift-ecosystem/operator-pipelines/main/fbc/Makefile
```

2. Run the FBC onboarding command:

```
make fbc-onboarding
```

3. The script performs the following actions:

- a. Downloads required tools (**opm**, **fbc-onboarding CLI**)
- b. Fetches supported OpenShift catalogs
- c. Generates catalog templates
- d. Generates the FBC structure
- e. Updates the **ci.yaml** configuration file

#### Verification Steps

After successful onboarding, you should see the following structure:

#### Operator directory:

```
$ tree .
operators/aqua
├── 0.0.1
...
└── catalog-templates
    ├── v4.12.yaml
    ├── v4.13.yaml
    ├── v4.14.yaml
    ├── v4.15.yaml
    └── v4.16.yaml
└── ci.yaml
```

#### FBC catalog directory:

To view the catalog directory structure created by the FBC onboarding process, first navigate to the root of your forked repository:

```
$ cd ../../
```

Then run:

```
$ tree (repository root)/catalogs
```

This will display the structure under the **catalogs**/ directory. For example:

```
catalogs
├── v4.12
│   └── aqua
│       └── catalog.yaml
├── v4.13
│   └── aqua
│       └── catalog.yaml
├── v4.14
│   └── aqua
│       └── catalog.yaml
├── v4.15
│   └── aqua
│       └── catalog.yaml
└── v4.16
    └── aqua
        └── catalog.yaml
```

### 22.3.2. Submitting FBC changes

After completing the onboarding and validation, add the generated resources to your Git repository and submit them through a pull request.

```
$ git add operators/aqua/{catalog-templates,ci.yaml,Makefile}
```

```
$ git add catalogs/{v4.12,v4.13,v4.14,v4.15,v4.16}/aqua
$ git commit --signoff -m "Add FBC resources for aqua operator"
```

When you merge the pull request, the operator pipeline will validate and publish the FBC content to the appropriate OpenShift catalogs.

## 22.4. FILE-BASED CATALOG WORKFLOW

To use the FBC workflow, first convert your existing non-FBC operator by using the [onboarding](#) process or start directly if you are certifying a new operator. The FBC workflow provides a modular, template-driven structure for managing OpenShift catalogs and supports automatic updates through configuration in the **ci.yaml** file.

### 22.4.1. Enabling FBC in the workflow

To use the FBC workflow, update the **ci.yaml** file in your operator repository by adding the following field:

```
fbc:
  enabled: true
```

This tells the certification pipeline to use the FBC workflow to generate and update catalogs.

### 22.4.2. FBC templates

File-Based catalog templates provide a simplified, user-editable view of an OpenShift catalog. The Operator Package Manager (OPM) currently supports two types of templates. You can select the template that best fits your operator's release strategy:

- **olm.template.basic**: A straightforward template for simple upgrade flows.
- **olm.semver**: A more advanced template that supports automatic channel generation by using semantic versioning.

Red Hat recommends using **olm.semver** for improved automation.

For details about the template schema, see [Operator Framework documentation](#).

### 22.4.3. Mapping templates to OpenShift catalogs

To generate catalogs from templates, you need to provide a mapping between the template and the catalog in the **ci.yaml** file. Depending on your needs, you can map a template either with **1:N** mapping or **1:1** mapping

#### 1:N mapping example

```
---
fbc:
  enabled: true
  catalog_mapping:
    - template_name: my-custom-semver-template.yaml # The name of the file inside ./catalog-templates directory
      catalogs_names: # a list of catalogs within the /catalogs directory
```

```

    - "v4.15"
    - "v4.16"
    - "v4.17"
  type: olm.semver
- template_name: my-custom-basic-template.yaml # The name of the file inside catalog-templates
  directory
  catalogs_names:
    - "v4.12"
    - "v4.13"
  type: olm.template.basic

```

### 1:1 mapping example

```

---
fbc:
  enabled: true
  catalog_mapping:
    - template_name: v4.14.yaml
      catalog_names: ["v4.14"]
      type: olm.template.basic
    - template_name: v4.15.yaml
      catalog_names: ["v4.15"]
      type: olm.template.basic
    - template_name: v4.16.yaml
      catalog_names: ["v4.16"]
      type: olm.template.basic
    - template_name: v4.17.yaml
      catalog_names: ["v4.17"]
      type: olm.template.basic

```

#### 22.4.4. Generating catalogs from templates

Use the provided **Makefile** to simplify catalog generation from templates. It automates the process by running the appropriate **opm** commands for each template type.

If you followed the [onboarding](#) process for a new or converted operator, your operator repository should already include the **Makefile**.

Make sure it is available in the root directory of your operator component:

```

  └── 0.0.1/
    ├── release-config.yaml
    └── manifests/
      └── metadata/
    └── catalog-templates/
    └── ci.yaml
  └── Makefile

```

Run the following command to generate FBC catalog files for each supported OpenShift version:

```
make catalogs
```

This command processes your catalog templates and creates structured catalog files in the **catalogs/** directory. After generation, submit the updated catalog files through a pull request. When you merge the changes, the operator pipeline publishes the updates to the OpenShift Container Platform index.

#### Example output structure:

```

catalogs/
  └── v4.12/
      └── aqua/
          └── catalog.yaml
  └── v4.13/
      └── aqua/
          └── catalog.yaml
  └── v4.14/
      └── aqua/
          └── catalog.yaml
  └── v4.15/
      └── aqua/
          └── catalog.yaml
  └── v4.16/
      └── aqua/
          └── catalog.yaml

```

### 22.4.5. Adding a new bundle to the catalog

Add a new operator bundle to the catalog by using two methods:

- Automatic process
- Manual process

#### 22.4.5.1. Using Automated Release

Automated release streamlines the bundle release and catalog update process.

Use this approach to reduce manual intervention and maintain consistency across operator versions. For details on enabling and using this feature, see [File-based Catalog auto-release](#).

#### 22.4.5.2. Manually adding bundle

Follow the steps to add operator bundles manually or by using a basic template:

1. Submit the new operator version using the traditional [pull request \(PR\) workflow](#).
2. The operator pipeline builds, tests, and releases the bundle image to the registry.
3. After releasing the bundle, update your catalog templates with the new bundle image **pullspec**. You can find this **pullspec** in the comment left by the pipeline on the PR.
4. Create a new PR that includes the catalog updates referencing the newly released bundle.



#### NOTE

Manual updates require a two-step PR process—one to release the bundle and another to update the catalog templates.

## Additional resources

- For guidance on authoring and editing catalog templates (such as SemVer or basic schemas), see [OPM documentation](#).

### 22.4.6. Updating existing catalogs

One of the key advantages of the File-Based Catalog format is that it allows you to modify an operator's update graph after releasing the bundle. This flexibility enables you to:

- Adjust the order of version updates
- Remove outdated or invalid bundles
- Apply other post-release changes without re-releasing a bundle

After updating the catalog templates, run **make catalogs** to generate the updated catalog. Then, submit the generated files through the standard pull request workflow.



#### NOTE

Submit all the catalog changes by using the standard pull request workflow for all the updates to take effect.

## 22.5. FILE-BASED CATALOG AUTO-RELEASE

Auto-release simplifies the FBC workflow by automating catalog updates after releasing a bundle. This reduces manual effort and helps ensure consistency across your catalog templates.

### 22.5.1. Overview

The standard FBC release workflow includes two steps:

- Build, test, and release the operator bundle
- Add the released bundle to the OpenShift Container Platform (OCP) catalog

You can now automate the second step by using the auto-release feature. Enable the auto-release feature to automatically update the catalog.

After you submit a pull request that includes both the new bundle and the **release-config.yaml** file, the release pipeline takes care of the rest. It creates and merges a follow-up pull request that contains the necessary catalog changes. [Here's](#) an example of an auto-generated catalog PR. ".

### 22.5.2. Creating the release-config.yaml file

To enable the automatic release of your operator bundle to OpenShift Container Platform (OCP) catalogs in FBC mode, add a **release-config.yaml** file in the corresponding bundle version directory. For example, **operators/aqua/0.0.2/release-config.yaml**.

```
tree operators/aqua
.
└── 0.0.2
    └── release-config.yaml # This is the file
```

```

  |
  +-- manifests
  |   +-- metadata
  +-- catalog-templates
  +-- ci.yaml
  +-- Makefile

```

The **release-config.yaml** file defines where the new bundle will be released—specifically, which catalog templates to update and how the bundle fits into the update graph.

### Example

```

---
catalog_templates:
  - template_name: basic.yaml
    channels: [my-channel]
    replaces: aqua.0.0.1
  - template_name: semver.yaml
    channels: [Fast, Stable]

```

In this example:

- The operator bundle is released to the **my-channel** channel in the **basic.yaml** template.
- The same bundle is also released to the **Fast** and **Stable** channels in the **semver.yaml** template.
- The **replaces** field (optional) specifies the earlier bundle version that the new bundle supersedes in the update graph.

### 22.5.3. File structure details

The structure of the **release-config.yaml** file is validated automatically during the pipeline run. If the file does not conform to the schema, the pull request will fail with a detailed error message.

The file must use the following structure:

- The top-level key is **catalog\_templates**, which contains a list of catalog update definitions.
- Each item in the list represents one catalog template and must include:
  - **template\_name**: Name of the catalog template file located in the **catalog-templates/** directory.
  - **channels**: List of channels to which the bundle should be released.  
For SemVer templates, valid values are **Fast**, **Stable**, and **Candidate**.
  - **replaces** (optional): Specifies which bundle the new one replaces in the update graph. This is valid only for the basic templates.
  - **skips** (optional): A list of bundles to skip in the update graph. This is valid only for the basic templates.
  - **skipRange** (optional): A range indicating which bundle versions should be skipped. This is valid only for the basic templates.

For more information, see [release-config.yaml schema documentation](#).

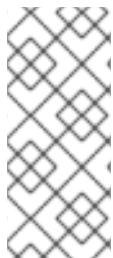
## 22.6. CREATING A PULL REQUEST

The final step is to create a pull request for the targeted upstream repo.

Catalog	Upstream Repository
Certified Catalog	<a href="https://github.com/redhat-openshift-ecosystem/certified-operators">https://github.com/redhat-openshift-ecosystem/certified-operators</a>

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.

### 22.6.1. 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

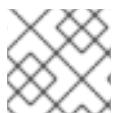
Red Hat OpenShift software certification or validation does not conduct testing of the Partner's product in how it functions or performs outside of the Operator constructs and its impact on the Red Hat platform on which it was installed and executed. Any and all aspects of the certification candidate 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. RUNNING THE CERTIFICATION TEST SUITE LOCALLY

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



### NOTE

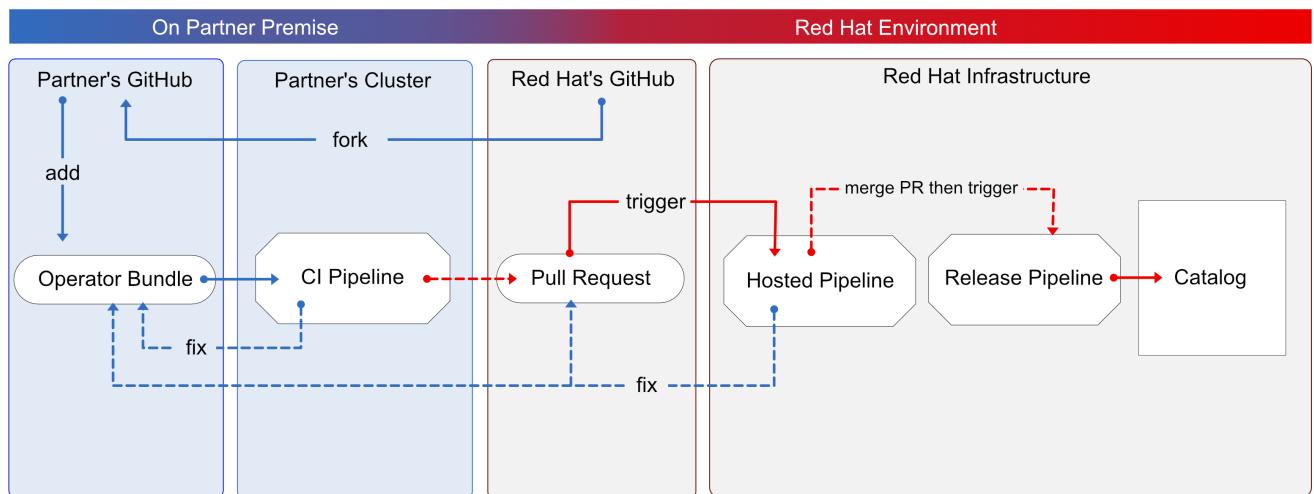
Red Hat recommends you to follow this method to certify your operators.

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 B.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](#).

**B.1. ADDING YOUR OPERATOR BUNDLE**

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

**NOTE**

If you want to ship your operator by using the File-Based Catalog (FBC) workflow, see [File-based Catalog \(FBC\)](#).

**B.1.1. If you have certified this operator before -**

Find your operator folder 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.

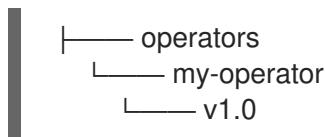
**B.1.2. If you are newly certifying this operator -**

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

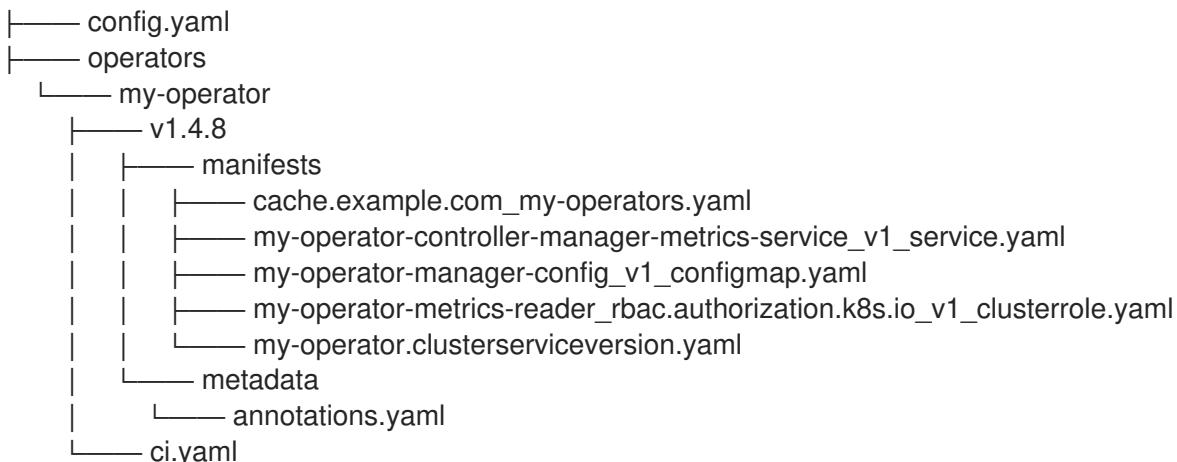
example, <**my-operator**> and create a version directory for example, <**V1.0**> and place your bundle. The certification process preinstalls these directories for operators that were previously certified.



- 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.



Configuration file	Description
config.yaml	In this file include the organization of your operator. It can be <b>certified-operators</b> . For example, <b>organization: certified-operators</b>
ci.yaml	In this file include your Red Hat Technology Partner Component PID for this operator.  For example, <b>cert_project_id: &lt;your component pid&gt;</b> . This file stores all the necessary metadata for a successful certification process.

Configuration file	Description
annotations.yaml	<p>In this file include an annotation of OpenShift versions, which refers to the range of OpenShift versions . For example, <b>v4.8-v4.10</b> means versions 4.8 through 4.10. Add this to any existing content.</p> <p>For example, <b># OpenShift annotations</b>  <b>com.redhat.openshift.versions: v4.8-v4.10</b>.</p> <p>The <b>com.redhat.openshift.versions</b> 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.</p> <p>Note that the letter 'v' must be used before the version, and spaces are not allowed. The syntax is as follows:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• A single version preceded by '=' indicates that the operator is supported only on that specific version of OpenShift. For example, using <b>=v4.8</b> will add the operator to the certified catalog for OpenShift 4.8, but not for later OpenShift releases.</li> <li>• A range can be used to indicate support only for OpenShift versions within that range. For example, using <b>v4.8-v4.10</b> will add the operator to the certified catalog for OpenShift 4.8 through 4.10, but not for OpenShift 4.11 or 4.12.</li> </ul>

## Additional resources

- [An example of an operator bundle](#)

## B.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:

Catalog	Upstream Repository
Certified Catalog	<a href="https://github.com/redhat-openshift-ecosystem/certified-operators">https://github.com/redhat-openshift-ecosystem/certified-operators</a>

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](#).

## B.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](#)

### B.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.

### B.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:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: my-local-pv
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  local:
```

```

path: /dev/vda4 ← use a path from your cluster
nodeAffinity:
  required:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - crc-8k6jw-master-0 ← use the name of one of your cluster's node

```

- 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](#).

### B.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:
 

```
oc get pods -n openshift-marketplace
```
  - Record the full pod name of the **certification-operator-controller-manager** pod and run the command:
 

```
oc get logs -f -n openshift-operators <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**.

### B.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](#).

## B.3.2. Manual process

Follow these steps to manually install the OpenShift Operator Pipeline:

### B.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**.

### B.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:

```
export KUBECONFIG=/path/to/your/cluster/kubeconfig
```

The **kubeconfig** file deploys the Operator under test and runs the certification checks.

### Additional resources

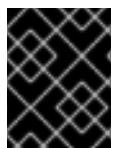
For more information about kubeconfig files, see [Organizing Cluster Access Using kubeconfig Files](#).

#### B.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.

#### B.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 about the kubeconfig secret, see [Secrets](#).

#### B.3.2.5. Importing Operator from Red Hat Catalog

Import Operators from the [Red Hat catalog](#).

In a terminal window, run the following commands:



#### NOTE

Before running the commands, identify the major and minor versions of your OpenShift cluster (for example, OPENSHIFT\_VERSION=v4.19)

```
oc import-image certified-operator-index:${OPENSHIFT_VERSION} \
--from=registry.redhat.io/redhat/certified-operator-index:${OPENSHIFT_VERSION} \
--reference-policy local \
--scheduled \
--confirm
oc import-image redhat-marketplace-index:${OPENSHIFT_VERSION} \
--from=registry.redhat.io/redhat/redhat-marketplace-index:${OPENSHIFT_VERSION} \
```

```
--reference-policy local \
--scheduled \
--confirm
```

**NOTE**

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

```
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.

### B.3.2.6. Installing the certification pipeline dependencies

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

```
$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
```

### B.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:

#### B.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:

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

#### B.3.2.7.2. Adding Red Hat Container API access key

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

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

#### B.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
```

#### B.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>
```

## B.4. EXECUTE THE OPENSHIFT 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.yaml
```

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 about SCCs, see [About security context constraints](#).

**B.4.1. Running the Minimal pipeline****Procedure**

In a terminal window, run the following commands:

```
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.

```
--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.

### B.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:

```
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**.

### B.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:

```
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 \
--workspace name=registry-credentials,secret=registry-docker config-secret \
--showlog \
```

## B.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:

### B.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 \
--param submit=true \
--param env=prod \
--workspace name=pipeline,volumeClaimTemplateFile=templates/workspace-template.yml \
--showlog
```

### B.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=redhat-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**.

### B.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=redhat-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
```

### B.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 \
--param pin_digests=true \
--param git_username=$GIT_USERNAME \
--param git_email=$GIT_EMAIL \
--param upstream_repo_name=redhat-openshift-ecosystem/certified-operators \
--param registry=$REGISTRY \
--param image_namespace=$IMAGE_NAMESPACE \
--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
```

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](#).

## APPENDIX C. HELM AND ANSIBLE OPERATORS

- For information about building a Helm operator, see [Building a Helm Operator](#).
- For information about building an Ansible operator, see [Building an Ansible Operator](#).

## PART IV. CERTIFYING OR VALIDATING HELM CHARTS

# CHAPTER 25. WORKING WITH HELM CHARTS



## NOTE

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

## 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.

**Partner Validation** - Select this type of certification, if you want to validate your product using your own criteria and test suite on Red Hat platforms. This partner validation allows you to publish your software offerings on the Red Hat Ecosystem Catalog more quickly. However, validated workloads may not incorporate all of Red Hat integration requirements and best practices. We encourage you to continue your efforts toward Red Hat certification.

**Certified** - Select this type of certification, if you want your product to undergo thorough testing by using Red Hat's test suite, and benefit from collaborative support. Your products will meet your standards and Red Hat's criteria, including interoperability, lifecycle management, security, and support requirements.

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

It is critical for partners to understand the distinction between Partner Validation and Red Hat Certification. Find below a clear differentiation between Validation and Certification programs:

Feature	Partner Validation	Red Hat Certification
Testing Criteria	Determined by Partner	Determined by both Red Hat and Partner
Testing Tools	Provided by Partner	Provided by both Red Hat and Partner
Red Hat Review	Partner statements, supportive documentation and test results	Test results, partner statements, and documentation
Scope	Quicker market entry, initial presence on the Red Hat Ecosystem Catalog	Comprehensive compliance with Red Hat platform requirements, best practices such as interoperability, security, and support

## Additional resources

- [Introduction to Partner Validation](#)

## 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.

### Task Summary

The certification workflow includes three primary steps-

1. [Section 25.2.1, "Certification on-boarding for Helm charts"](#)
2. [Section 25.2.2, "Certification testing for Helm charts"](#)
3. [Section 25.2.3, "Publishing the certified Helm chart on the Red Hat Ecosystem Catalog"](#)

### 25.2.1. Certification on-boarding for Helm charts

Perform the steps outlined for certification onboarding:

1. Join the [Red Hat Connect](#) for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Create your product listing by selecting your desired product category. You can select from the available product categories:
  - a. Containerized Application
  - b. Standalone Application
  - c. OpenStack Infrastructure
4. Complete your company profile.
5. Add components to the product listing.
6. Certify components for your product listing.

### Additional resources

For detailed instructions about creating your first product listing, see [Creating a product](#).

### 25.2.2. Certification testing for Helm charts

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

The **Certified** Helm charts must be added 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.

#### 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. VERIFYING HELM CHARTS FOR CERTIFICATION

You can verify 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 HatOpenShift 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

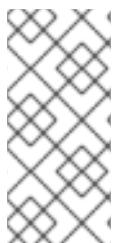
Red Hat strongly recommends using the latest version of the chart-verifier tool to validate your Helm charts on your local test environment. This enables you to check the results on your own during the chart development cycle, preventing the need to submit the results to Red Hat every time.

### 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 [Red Hat Partner Subscriptions](#).

### Procedure

- Set up a test server with x86-64 based Red Hat Enterprise Linux system consisting of OpenShift command line interface (oc), Helm, and either Podman or the chart-verifier tool installed.
- 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.
- Alternatively, install a self-managed cluster on your cloud environment, datacenter or computer. By using this option you can use your partner subscriptions, also known as NFRs, for permanent deployments.

### Additional resources

- For more information about setting up your environment, see [Try Red Hat OpenShift](#).
- 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

- 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.

- 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.

- 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
connection
--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:

```
$ 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:

```
$ 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.

- 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>
```

- 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>
```

### 26.2.1.1. 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 \
"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/chartverifier**.

For example:

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

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 \
<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 zxvf <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. CREATE A PRODUCT

The product listing provides marketing and technical information, showcasing your product's features and advantages to potential customers. It lays the foundation for adding all necessary components to your product for certification.

## 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 before certification.

Certify your chart's container images as a container application before creating a Helm chart component.

## Procedure

Red Hat recommends completing all optional fields in the listing tabs for a comprehensive product listing. More information helps mutual customers make informed choices.

Red Hat encourages collaboration with your product manager, marketing representative, or other product experts when entering information for your product listing.

Fields marked with an asterisk (\*) are mandatory.

## Procedure

1. Log in to the [Red Hat Partner Connect Portal](#).
2. Go to the Certified technology portal tab and click **Visit the portal**.
3. On the header bar, click **Product management**.
4. From the **Listing and certification** tab click **Manage products**.
5. From the **My Products** page, click **Create Product**.  
A **Create New Product** dialog opens.
6. Enter the **Product name**.
7. From the **What kind of product would you like to certify?** drop-down, select the required product category and click **Create product**. For example, select **Containerized Application** for creating a containerized product listing.

A new page with your Product name opens. It comprises the following tabs:

- [Section 27.1, "Overview for Helm charts"](#)
- [Section 27.2, "Product Information for Helm charts"](#)
- [Section 27.3, "Components for Helm charts"](#)
- [Section 27.4, "Support for Helm charts"](#)

Along with the following tabs, the page header provides the **Product Score** details. Product Score evaluates your product information and displays a score. It can be:

- Fair

- Good
  - Excellent
  - Best
8. Click **How do I improve my score?** to improve your product score.
  9. After providing the product listing details, click **Save** before moving to the next section.

## 27.1. OVERVIEW FOR HELM CHARTS

This tab consists of a series of tasks that you must complete to publish your product:

- [Section 27.1.1, "Complete product listing details for Helm charts"](#)
- [Section 27.1.2, "Complete company profile information for Helm charts"](#)
- [Section 27.1.3, "Accept legal agreements for Helm charts"](#)
- [Section 27.1.4, "Certify your Helm charts"](#)
- [Section 27.1.5, "Add at least one product component for Helm charts"](#)
- [Section 27.1.6, "Certify components for your listing for Helm charts"](#)

### 27.1.1. Complete product listing details for Helm charts

1. To complete your product listing details, click **Start**.  
The **Product Information** tab opens.
2. Enter all the essential product details and click **Save**.

### 27.1.2. Complete company profile information for Helm charts

1. To complete your company profile information, click **Start**. After entering all the details, click **Submit**.
2. To modify the existing details, click **Review**. The **Account Details** page opens.
3. Review and modify the Company profile information and click **Submit**.

### 27.1.3. Accept legal agreements for Helm charts

To publish your product image, agree to the terms regarding the distribution of partner container images.

1. To accept the legal agreements, click **Start**.
2. To preview or download the agreement, click **Review**.

The **Red Hat Partner Connect Container Appendix** document displays. Read the document to know the terms related to the distribution of container images.

### 27.1.4. Certify your Helm charts

1. Click **Certify product**. Read the [Publication and testing guidelines](#)
  - a. Click **Add Component** and then go to [Section 27.1.5, "Add at least one product component for Helm charts"](#).

## 27.1.5. Add at least one product component for Helm charts

1. Click **Start**. You are redirected to the **Components** tab.  
To add a new or existing product component, click **Add component**.
2. For adding a new component,
  - a. In the **Component Name** text box, enter the component name.
  - b. For **What kind of standalone component are you creating?** select the component that you wish to certify. For example, for certifying your Helm Charts, select **Helm Chart**.
  - c. Click **Next**.
  - d. In the **Chart Name** text box, enter a unique name for your chart.
  - e. **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 component 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.
- f. Click **Add component**.
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

## 27.1.6. Certify components for your listing for Helm charts

1. To certify the components for your listing, click **Start**. If you have existing product components, you can view the list of **Attached Components** and their details:
  - a. Name
  - b. Certification
  - c. Security
  - d. Type
  - e. Created
  - f. Click more options to archive or remove the components
2. Select the components for certification.

After completing all the above tasks you will see a green tick mark corresponding to all the options.

The Overview tab also provides the following information:

1. **Product contacts** - Provides Product marketing and Technical contact information.
  - a. Click **Add contacts to product** to provide the contact information
  - b. Click **Edit** to update the information.
2. **Components in product** - Provides the list of the components attached to the product along with their last updated information.
  - a. Click **Add components to product** to add new or existing components to your product.
  - b. Click **Edit components** to update the existing component information.

After publishing the product listing, you can view your **Product Readiness Score** and **Ways to raise your score** on the **Overview** tab.

#### Additional resources

For more information about the distribution methods, see [Helm Chart Distribution methods](#).

## 27.2. PRODUCT INFORMATION FOR HELM CHARTS

Through this tab you can provide all the essential information about your product. The product details are published along with your product on the Red Hat Ecosystem catalog.

**General** tab:

Provide 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. Click **Save**.

**Features & Benefits tab:**

Provide important features of your product.

1. Optional: Enter the **Title** and **Description**.
2. Optional: To add additional features for your product, click + **Add new feature**.
3. Click **Save**.

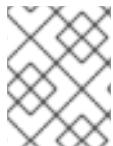
**Quick start & Config tab:**

Add links to any quick start guide or configuration document to help customers deploy and start using your product.

1. Optional: Enter **Quick start & configuration instructions**.
2. Click **Save**.
3. Select **Hide default instructions** check box, if you don't want to display them.

**Linked resources tab:**

Add links to supporting documentation to help our customers use your product. The information is mapped to and is displayed in the Documentation section on the product's catalog page.

**NOTE**

It is mandatory to add a minimum of three resources. Red Hat encourages you to add more resources, if available.

1. Select the **Type** drop-down menu, and enter the **Title** and **Description** of the resource.
2. Enter the **Resource URL**.
3. Optional: To add additional resources for your product, click + **Add new Resource**.
4. Click **Save**.

**FAQs tab:**

Add frequently asked questions and answers of the product's purpose, operation, installation, or other attribute details. You can include common customer queries about your product and services.

1. Enter **Question** and **Answer**.
2. Optional: To add additional FAQs for your product, click + **Add new FAQ**.
3. Click **Save**.

**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**.

#### Contacts tab:

Provide contact information of your marketing and technical team.

1. Enter the **Marketing contact email address** and **Technical contact email address**.
2. Optional: To add additional contacts, click **+ Add another**.
3. Click **Save**.

#### Legal tab:

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:

Use this tab to improve the 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.
3. Click **Save**.
4. Optional: To add additional key-value pair, click **+ Add new key-value pair**.



#### NOTE

Add at least one Search alias for your product. Red Hat encourages you to add more aliases, if available.

### 27.3. COMPONENTS FOR HELM CHARTS

Use this tab to add components to your product listing. Through this tab you can also view a list of attached components linked to your Product Listing.

Alternatively, to attach a component to the Product Listing, you can complete the **Add at least one product component** option available in the **Overview** tab of a Container, Operator, or Helm Chart product listing.

1. To add a new or existing product component, click **Add component**.
2. For adding a new component,
  - a. In the **Component Name** text box, enter the component name.
  - b. For **What kind of OpenShift component are you creating?** select the component that you wish to certify. For example, for certifying your Helm Charts, select **Helm Chart**.

- c. Click **Next**.
- d. In the **Chart Name** text box, enter a unique name for your chart.
- e. **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 in the Red Hat OpenShift Developer Console. This is the default option when you create a new component 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.
  - f. Click **Add component**.
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
- a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.



#### NOTE

You can add the same component to multiple products listings. All attached components must be published before the product listing can be published.

After attaching components, you can view the list of **Attached Components** and their details:

- i. Name
- ii. Certification
- iii. Security
- iv. Type
- v. Created
- vi. Click more options to archive or remove the attached components

Alternatively, to search for specific components, type the component's name in the **Search by component Name** text box.

## 27.4. SUPPORT FOR HELM CHARTS

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

You can also contact the Red Hat Partner Acceleration Desk for any technical questions you may have regarding the Certification. Technical help requests will be redirected to the Certification Operations team.

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.

1. To request support, click Open a support case. See [PAD - How to open & manage PAD cases](#), to open a PAD ticket.
2. To view the list of existing support cases, click [View support cases](#).

## 27.5. REMOVING A PRODUCT

After creating a product listing if you wish to remove it, go to the **Overview** tab and click **Delete**.

A published product must first be unpublished before it can be deleted. Red Hat retains information related to deleted products even after you delete the product.

## CHAPTER 28. ADDING CERTIFICATION COMPONENTS

After creating the new product listing, add the certification components for the newly created product listing.

You can configure the following options for the newly added components:



### NOTE

The component configurations differ for different product categories.

- [Section 28.1, "Certification for Helm charts"](#)
- [Section 28.2, "Optional Qualifications for Helm charts"](#)
- [Section 28.3, "Repository information for Helm charts"](#)
- [Section 28.4, "Component details for Helm charts"](#)
- [Section 28.5, "Contact Information for Helm charts"](#)
- [Section 28.6, "Associated products for Helm charts"](#)

To configure the options, go to the **Components** tab and click on any of the existing components.

### 28.1. CERTIFICATION FOR HELM CHARTS

#### GitHub Verification

After creating your Helm Chart component on Red Hat partner connect, submit your Helm Chart for verification.

1. From the **Certification** tab, go to [GitHub Verification](#).
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 more information about submitting your pull request, see [Submitting your Helm chart for certification](#).

### 28.2. OPTIONAL QUALIFICATIONS FOR HELM CHARTS



### NOTE

This tab is applicable only for Operator and Helm chart certifications.

Use **Optional qualifications** tab to verify if your product follows Red Hat's recommended guidelines and

best practices for deploying workload on Red Hat OpenShift. When you select this tab, a functional certification is created where you will submit testing results for Red Hat's review. After successful verification, your workload product gets listed as Certified with the **Meets Best Practices** badge on the Red Hat Ecosystem catalog.

### Additional resources

For more information, see [Best practices](#)

## 28.3. REPOSITORY INFORMATION FOR HELM CHARTS

### Distribution via External Helm chart repository

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:

Field name	Description
Chart name	Unique name of your Helm chart
Container registry namespace	Registry name set when the container was created. This field becomes non-editable when the container gets published.
Helm chart repository	It 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	It denotes the GitHub users who are allowed to submit Helm charts for certification on behalf of your company.
Short and Long repository descriptions	This information will be used when listing your Helm chart on the Red Hat Ecosystem Catalog.

After configuring all the mandatory fields click **Save**.



### NOTE

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

## 28.4. COMPONENT DETAILS FOR HELM CHARTS

Configure the product component details by using this tab.

Enter the required details in the following fields:

Field name	Description
Application categories	Select the respective application type of your software product.
Component name	Name of the component for internal purposes.

After configuring all the mandatory fields click **Save**.

## 28.5. CONTACT INFORMATION FOR HELM CHARTS



### NOTE

Providing information for this tab is optional.

In the **Contact Information** tab, enter the primary technical contact details of your product component.

1. Optional: In the **Technical contact email address** field, enter the email address of the image maintainer.
2. Optional: To add additional contacts for your component, click **+ Add new contact**
3. Click **Save**.

## 28.6. ASSOCIATED PRODUCTS FOR HELM CHARTS

The Associated Product tab provides the list of products that are associated with your product component along with the following information:

- Product Name
- Type – Traditional application
- Visibility – Published or Not Published
- Last Activity – number of days before you ran the test

To add products to your component, perform the following:

- If you want to find a product by its name, enter the product name in the **Search by name** text box and click the search icon.
- If you are not sure of the product name, click **Find a product**. From the **Add product** dialog, select the required product from the Available products list box and click the forward arrow. The selected product is added to the Chosen products list box. Click **Update attached products**. Added products are listed in the Associated product list.

**NOTE**

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

# CHAPTER 29. SUBMITTING YOUR HELM CHART FOR CERTIFICATION

After configuring and setting up your Helm chart component 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 component 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 component, 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 components, 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**

```
charts/partners/acme/awesome/0.1.0/awesome-0.1.0.tgz
charts/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
```

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

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

```
.  
└── 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
```

## 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,

```
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.

Objective	Include Helm chart	Include chart verification report	Red Hat certification outcome	Methods to publish your certified Helm chart
If you want to perform the following actions: <ul style="list-style-type: none"> <li>Store your certified chart at <b>charts.openshift.io</b>.</li> <li>Take advantage of Red Hat CI for ongoing chart tests</li> </ul>	Yes	No	The <a href="#">chart-verifier tool</a> is executed in the Red Hat CI environment to ensure compliance.	Your customers can download the certified Helm charts from <b>charts.openshift.io</b> .
If you want to perform the following actions: <ul style="list-style-type: none"> <li>Store your certified chart at <b>charts.openshift.io</b>.</li> <li>Aim to test your chart in your own environment since it has some external dependencies.</li> </ul>	Yes	Yes	The Red Hat certification team reviews the results to ensure compliance.	Your customers can download the certified Helm charts from <b>charts.openshift.io</b> .

Objective	Include Helm chart	Include chart verification report	Red Hat certification outcome	Methods to publish your certified Helm chart
If you don't want to store your certified charts at <b>charts.openshift.io</b> .	No	Yes	The Red Hat certification team reviews the results to ensure compliance.	Your customers can download the certified Helm chart from your designated Helm chart repository. A corresponding entry is added to the <b>index.yaml</b> file at <b>charts.openshift.io</b> .

## 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 verification through a pull request, the Red Hat certification team reviews and verifies your component for certification. After successful verification, your Helm chart is certified through GitHub.

Follow the steps to publish your certified Helm chart:

1. Access the [Partner connect](#) web page. **My Products** web page displays the **Product Listings**.
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. Go to the **Components** tab.
5. Click **Add component** and go to **Existing component** tab to attach your certified Helm chart to this listing. Also add the certified containers used by your Helm chart. Both the components 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 components.
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: VIRTUALIZATION, BEST PRACTICES, CNF, CNI, CSI

Red Hat OpenShift certification badges extend the Red Hat OpenShift Operator certification. The certification badges are built on the foundation of container and operator certifications.

By receiving a Red Hat OpenShift Certification Badge, partners can confirm that their solution is Kubernetes enabled, meets Kubernetes best practices and utilizes specific Kubernetes APIs for addressing the respective use cases.

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

- [OpenShift Virtualization](#)—for providing support to run and manage virtual machine workloads on Red Hat OpenShift.
- [Meets Best practices](#)—for meeting the Red Hat best practices checkpoints for cloud native software products that are deployed on Red Hat OpenShift.
- [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. OPENSHIFT VIRTUALIZATION VALIDATION

## 31.1. VALIDATING YOUR PRODUCTS WITH THE OPENSIFT VIRTUALIZATION FEATURE

Red Hat OpenShift Virtualization is an add-on to OpenShift Container Platform. It provides the capability to run and deploy your new and existing virtual machine (VM) workloads. The solution allows for easy migration and management of traditional virtual machines onto a trusted, consistent, and comprehensive hybrid cloud application platform.

The OpenShift Virtualization badge is a specialization within Red Hat OpenShift certification. Products that meet the requirements and complete the workflow get listed on the [Red Hat Ecosystem Catalog](#).

### Additional resources

For more information about OpenShift Virtualization, see:

- [Red Hat OpenShift Virtualization](#)
- [Migrate virtual machines to Red Hat OpenShift](#)
- [What you can do with OpenShift Virtualization](#)

## 31.2. CREATING A PRODUCT

The product listing provides marketing and technical information, showcasing your product's features and advantages to potential customers. It lays the foundation for adding all necessary components to your product for validation.

### Prerequisites

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

### Procedure

Red Hat recommends completing all optional fields in the listing tabs for a comprehensive product listing. The more information you provide helps mutual customers make informed choices.

Red Hat encourages collaboration with your product manager, marketing representative, or other product experts when entering information for your product listing.

Fields marked with an asterisk (\*) are mandatory.

1. Log in to the [Red Hat Partner Connect Portal](#).
2. From the **Activities > Software certification** tile, click **Certify your software**.
3. On the header bar, click **Product management**.
4. From the **Listing and certification** tab click **Manage products**.
5. From the **My Products** page, click **Create Product**.  
A **Create New Product** dialog box opens.

6. Enter the **Product name**.
7. From the **What kind of product would you like to certify?** drop-down list, select the required product category and click **Create product**. For example, select **Containerized Application** or **Standalone Application** for creating a containerized or standalone product listing.

A new page with your Product name opens. It comprises the following tabs:

- [Section 31.2.1, "Overview for OpenShift Virtualization"](#)
- [Section 31.2.2, "Product Information for OpenShift Virtualization"](#)
- [Section 31.2.3, "Components for OpenShift Virtualization"](#)
- [Section 31.2.4, "Support for OpenShift Virtualization"](#)

Along with the following tabs, the page header provides the **Product Score** details. Product Score evaluates your product information and displays a score. The score can be:

- Fair
- Good
- Excellent
- Best

Click **How do I improve my score?** to improve your product score.

8. After providing the product listing details, click **Save** before moving to the next section.

### 31.2.1. Overview for OpenShift Virtualization

This tab consists of a series of tasks that you must complete to publish your product:

- [Section 6.1.1, "Complete product listing details"](#)
- [Section 6.1.2, "Complete company profile information"](#)
- [Section 31.2.1.3, "Accept legal agreements"](#)
- [Section 6.1.4, "Add at least one product component"](#)
- [Section 6.1.5, "Certify components for your listing"](#)

#### 31.2.1.1. Complete product listing details

1. To complete your product listing details, click **Start**.  
The **Product Information** tab opens.
2. Enter all the essential product details and click **Save**.

#### 31.2.1.2. Complete company profile information

1. To complete your company profile information, click **Start**. After entering all the details, click **Submit**.
2. To modify the existing details, click **Review**. The **Account Details** page opens.

3. Review and modify the Company profile information and click **Submit**.

### 31.2.1.3. Accept legal agreements



#### NOTE

This option is applicable only for containerized applications.

To publish your product image, agree to the terms regarding the distribution of partner container images.

1. To accept the legal agreements, click **Start**.
2. To preview or download the agreement, click **Review**.

The **Red Hat Partner Connect Container Appendix** document displays. Read the document to know the terms related to the distribution of container images.

### 31.2.1.4. Add at least one product component

1. Click **Start**. You are redirected to the **Components** tab.  
To add a new or existing product component, click **Add component**.
2. For creating a new containerized application component:
  - a. In the **Component Name** text box, enter the component name. This name is not published and is only for internal use.
  - b. For **What kind of OpenShift component are you creating?** select the component that you wish to validate. For example, select **OpenShift Virtualization**.
  - c. Click **Create new component**
3. For creating a new standalone application component:
  - a. In the **Component Name** text box, enter the component name. This name is not published and is only for internal use.
  - b. For **What kind of OpenShift component are you creating?** select between two component options - **Non containerized application** or **OpenShift Virtualization**.
    - i. **Non containerized application** - Select this option to certify a non-containerized application component on the Red Hat Enterprise Linux platform.  
**Red Hat Enterprise Linux Version**- Specifies the Red Hat Enterprise Linux version on which you want to validate your non-containerized application component. Select the required RHEL version.



#### NOTE

You can not change the RHEL version after creating the component.

- ii. **OpenShift Virtualization** - Select this option to validate a specialized application component on the Red Hat OpenShift Virtualization feature.

- c. Click **Create new component**

4. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
  - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.

### 31.2.1.5. Certify components for your listing

1. To certify the components for your listing, click **Start**. If you have existing product components, you can view the list of **Attached Components** and their details:
  - a. Name
  - b. Certification
  - c. Security
  - d. Type
  - e. Created
  - f. Click more options to archive or remove the components
2. Select the components for certification.

After completing all the above tasks you will see a green tick mark corresponding to all the options.

The **Overview** tab also provides the following information:

1. **Product contacts** - Provides product marketing and technical contact information.
  - a. Click **Add contacts to product** to provide the contact information
  - b. Click **Edit** to update the information.
2. **Components in product** - Provides the list of the components attached to the product along with their last updated information.
  - a. Click **Add components to product** to add new or existing components to your product.
  - b. Click **Edit components** to update the existing component information.

After publishing the product listing, you can view your **Product Readiness Score** and **Ways to raise your score** on the **Overview** tab.

### 31.2.2. Product Information for OpenShift Virtualization

Use this tab to provide all the essential information about your product. The product details are published along with your product on the Red Hat Ecosystem catalog.

**General** tab:

Provide 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. Click **Save**.

#### Features & Benefits tab:

Provide important features of your product.

1. Optional: Enter the **Title** and **Description**.
2. Optional: To add additional features for your product, click + **Add new feature**.
3. Click **Save**.

#### Quick start & Config tab:

Add links to any quick start guide or configuration document to help customers deploy and start using your product.

1. Optional: Enter **Quick start & configuration instructions** and click **Save**.
2. Select **Hide default instructions** check box, if you don't want to display them.

#### Linked resources tab:

Add links to supporting documentation to help our customers use your product. The information is mapped to and is displayed in the Documentation section on the product's catalog page.



#### NOTE

It is mandatory to add a minimum of three resources. Red Hat encourages you to add more resources, if available.

1. Select the **Type** drop-down menu, and enter the **Title** and **Description** of the resource.
2. Enter the **Resource URL**.
3. Optional: To add additional resources for your product, click + **Add new Resource**.
4. Click **Save**.

#### FAQs tab:

Add frequently asked questions and answers of the product's purpose, operation, installation, or other attribute details. You can include common customer queries about your product and services.

1. Enter **Question** and **Answer**.
2. Optional: To add additional FAQs for your product, click + **Add new FAQ**.
3. Click **Save**.

#### Support tab:

1. Provide contact information of your support team:

- a. Optional: **Support description**,
  - b. **Support web site** – This field is mandatory.
  - c. Optional: **Support phone number**, and
  - d. Optional: **Support email address**.
2. Click **Save**.

#### **Contacts** tab:

Provide contact information of your marketing and technical team.

1. Enter the **Marketing contact email address** and **Technical contact email address**.
2. Optional: To add additional contacts, click + **Add another**.
3. Click **Save**.

#### **Legal** tab:

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:

Use this tab to improve the 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**.
3. Optional: To add additional key-value pair, click + **Add new key-value pair**.



#### **NOTE**

Add at least one Search alias for your product. Red Hat encourages you to add more aliases, if available.

### **31.2.3. Components for OpenShift Virtualization**

Use this tab to add components to your product listing. Through this tab you can also view a list of attached components linked to your Product Listing.

Alternatively, to attach a component to the product listing, you can complete the **Add at least one product component** option available in the **Overview** tab of a Container, Operator, or Helm Chart product listing.

1. To add a new or existing product component, click **Add component**.
2. For creating a new containerized application component:

- a. In the **Component Name** text box, enter the component name. This name is not published and is only for internal use.
  - b. For **What kind of OpenShift component are you creating?** select the component that you wish to validate. For example, select **OpenShift Virtualization**.
  - c. Click **Create new component**
3. For creating a new standalone application component:
- a. In the **Component Name** text box, enter the component name. This name is not published and is only for internal use.
  - b. For **What kind of OpenShift component are you creating?** select between two component options - **Non containerized application** or **OpenShift Virtualization**.
    - i. **Non containerized application** - Select this option to certify a non-containerized application component on the Red Hat Enterprise Linux platform.  
**Red Hat Enterprise Linux Version**- Specifies the Red Hat Enterprise Linux version on which you want to validate your non-containerized application component. Select the required RHEL version.



#### NOTE

You can not change the RHEL version after creating the component.

- ii. **OpenShift Virtualization** - Select this option to validate a specialized application component on the Red Hat OpenShift Virtualization feature.
  - c. Click **Create new component**
4. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
- a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
  - b. Click **Attach existing component**.



#### NOTE

You can add one component to multiple products listings. All attached components must be published before the product listing can be published.

After attaching components, you can view the list of **Attached Components** and their details:

- i. Name
- ii. Certification
- iii. Security
- iv. Type

- v. Created
- vi. Click more options to archive or remove the attached components

Alternatively, to search for specific components, type the component's name in the **Search by component Name** text box.

### 31.2.4. Support for OpenShift Virtualization

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

You can also contact the Red Hat Partner Acceleration Desk for any technical questions you may have regarding the Certification. Technical help requests will be redirected to the Certification Operations team.

Red Hat offers free, not-for-resale software subscriptions by using the Partner Subscriptions program, 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.

1. To request support, click Open a support case. See [PAD - How to open & manage PAD cases](#), to open a PAD ticket.
2. To view the list of existing support cases, click [View support cases](#).

### 31.2.5. Removing a product

After creating a product listing if you wish to remove it, go to the **Overview** tab and click **Delete**.

A published product must first be unpublished before it can be deleted. Red Hat retains information related to deleted products even after you delete the product.

## 31.3. ADDING CERTIFICATION COMPONENTS

After creating the new product listing, add the certification components for the newly created product listing. You can configure the following options for the newly added components:



### NOTE

The component configurations differ for different product categories.

- [Section 31.3.1, "Validation"](#)
- [Section 16.5, "Component details"](#)
- [Section 31.3.3, "Contact information"](#)
- [Section 7.4, "Associated products"](#)

To configure the component options, go to the **Components** tab and click on any of the existing components.

### 31.3.1. Validation

Validate the functionality of your product on Red Hat OpenShift by using the **Certification** tab.

#### OpenShift Virtualization Validation

By using this feature the Red Hat certification team checks if your product meets all the standards for partner validation.

1. In the OpenShift Virtualization Validation page, click **Start questionnaire**. The **Partner Statement of Support for OpenShift Virtualization Questionnaire** page displays.
2. Enter all your product and company information.
3. After completing all the details, click **Submit**.
4. To modify the existing details, click **Review**. The **Partner Statement of Support for OpenShift Virtualization Questionnaire** page displays, allowing you to review and modify the entered information.

After you click **Submit**, a new functional validation request is created. The Red Hat certification team will review and validate the entered details of the questionnaire. The certification team will follow up with you in case of a problem. After successful verification, a green check mark is displayed with the validate complete message.

After successful review and validation, your functional validation request will be approved, and the **Certification Level** field in the product listing will be set to **Partner Validated**.

### 31.3.2. Component details

Configure the product component details by using this tab.

Enter the required details in the following fields:

- **Component name** - Enter the component name. This name is not published and is only for internal use.
- Click **Save**.

### 31.3.3. Contact information



#### NOTE

Providing information for this tab is optional.

In the **Contact Information** tab, enter the primary technical contact details of your product component.

1. Optional: In the **Technical contact email address** field, enter the email address of the image maintainer.
2. Optional: To add additional contacts for your component, click **+ Add new contact**.
3. Click **Save**.

### 31.3.4. Associated products

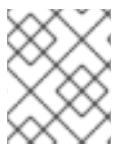
The **Associated Product** tab provides the list of product listings that are associated with your product component along with the following information:

- Product Name
- Type
- Visibility - Published or Not Published
- Last Activity - number of days before you ran the test

To create new or add an existing product listing to your component, perform the following steps:

- Perform the following steps for attaching an existing product listing:
  1. If you want to find a product by its name, enter the product name in the **Search by name** text box and click the search icon.
  2. If you are not sure of the product name, click **Find a product**. From the **Add product** dialog, select the required product from the Available products list box and click the forward arrow. The selected product is added to the Chosen products list box. Click **Update attached products**. Added products are listed in the Associated product tab.
- Perform the following steps for creating a new product listing:
  1. Click **Find product**.
  2. On the **Add Product** page, click **New Product**.
  3. In the **Product Name** text box, enter the new product name.
  4. From the **Product listing type** drop-down list, select the required product category. For example, select **Containerized Product or Standalone Application** for creating a containerized or standalone product listing.
  5. Click **Create new product**

Created product listing gets listed in the Associated product tab.



#### NOTE

All the fields marked with an asterisk \* are required and must be completed before you can proceed with the validation.

## 31.4. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit your OpenShift Virtualization component for validation, the Red Hat certification team will review and verify the entered details of the product questionnaire. After successful verification, go to the **Product Listings** page to publish your product on the [Red Hat Ecosystem Catalog](#).

Follow these steps to publish your product listing:

1. Access the [Red Hat Partner connect](#) web page. **My Work** web page displays the **Product Listings**.
2. Go 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. Go to the **Components** tab.
5. Click **Add Component** to add new component to your product listing.

**NOTE**

To publish your product listing on the [Red Hat Ecosystem Catalog](#), you must attach at least one published OpenShift Virtualization component to your product listing.

6. Click **Attach Component** to attach your OpenShift Virtualization component to this listing.
7. Click **Publish**.  
Your new OpenShift Virtualization product listing is now available for public access on the [Red Hat Ecosystem Catalog](#) along with the following details:
  - In the **Infrastructure features** section, the **OpenShift Virtualization** tag gets added.
  - In the **Certification Standards** section, you can see a validation confirmation message, for example - **This product has been validated by the partner**

# CHAPTER 32. MEETS BEST PRACTICES

## 32.1. MEETING BEST PRACTICES IN CLOUD NATIVE SOFTWARE CERTIFICATION

The **Meets Best Practices** badge is an optional specialization within Red Hat OpenShift workload certification that indicates your product follows the Red Hat best practices for containerized applications for the Red Hat OpenShift software certification.

These best practices comprises a series of checks that verifies whether the helm charts and operators that you submit for certification, meet the standard guidelines for deploying on Red Hat OpenShift. After successful verification your certified product gets listed with the **Meets Best Practices** badge on the [Red Hat Ecosystem catalog](#).



### NOTE

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

The certification workflow includes the following stages:

- Certification onboarding
- Creating a Product
- Adding components
- Certification testing
- Publishing the product listing on the Red Hat Ecosystem Catalog

## 32.2. CERTIFICATION ONBOARDING

1. Join the [Red Hat Partner Connect](#) for Technology Partner Program.
2. Agree to the [program terms and conditions](#).
3. Complete your [company profile](#).

For detailed information, see [Overview](#).

## 32.3. CREATING A PRODUCT

For detailed instructions about creating a product listing, see [Creating a product listing](#).

## 32.4. ADDING COMPONENTS

1. For Operators, see [Adding certification components for Operators](#).
2. For Helm charts, see [Adding certification components for Helm charts](#).

## 32.5. CERTIFICATION TESTING

## Procedure

1. Go to the **Components** tab > **Optional Qualifications**
2. Click **Start Testing**.
3. Click **Go to Red Hat certification tool** A new functional certification is created, after which you will be redirected to your component page on the [Red Hat Partner Certification](#) (rhcrt) portal.
4. Run the [Red Hat Best Practices Test Suite for Kubernetes](#) in your test environment with your product. It consists of a series of test cases derived from best practices established with our partners. The test suite will evaluate whether your product adheres to these principles and satisfies the Red Hat standards.
5. Perform the following steps on your product component page on the Red Hat Partner Certification (rhcrt) portal:
  - a. Go to the **Summary** tab:
    - i. From the **Files** section click **Upload**, to submit your product certification results. Select the **claims.json** and **tnf\_config.yml** files. And then, click **Next**. A successful upload message is displayed.
    - ii. Optional: Add your queries related to certification, if any, in the **Discussions** text box and then click **Add Comment**. The Red Hat certification team will provide clarifications for your queries.
  - b. Go to the **Properties** tab:
    - i. Click the **Platform** list menu to select the platform on which you want to certify your component. 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 component. For example, Red Hat OpenShift Platform
    - iii. Click **Update Values**. The selected values are updated.

## 32.6. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit your product for validation through the Red Hat certification portal , the Red Hat certification team reviews and verifies your product for certification. After successful verification, your certified product gets published on the Red Hat Ecosystem Catalog with the **Meets Best Practices** label.

## Procedure

1. Access the [Partner connect](#) web page. **My Products** web page displays the **Product Listings**.
2. Search for the newly created product listing that you wish to publish, click and review its details.
3. Go to the **Components** tab and click **Add component** to attach your certified operator or Helm chart to this listing. Also add any additional certified containers used by your product component. All the components must be in **Published** status. The Publish button is enabled when you complete all the required information for the product listing along with the attached product components.

4. Click **Publish**.

Your certified product is now available for public access on the [Red Hat Ecosystem Catalog](#).

The certified product will also be listed in the OperatorHub within the web console in OpenShift.

Partners will receive **Meets Best Practices badge** to promote their certified product on the Red Hat OpenShift platform.



**NOTE**

The Red Hat OpenShift software certification tests do not conduct functional testing of your product outside the Operator and Helm chart constructs. Also, it does not test your product's 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 remain the Partner's sole responsibility.

# CHAPTER 33. CNF CERTIFICATION AND VALIDATION

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

### 33.1.1. 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:

- **Partner Validation** - Select this type of CNF certification, if your container base image is neither RHEL nor UBI. For this type of certification, Partner Validate your CNF product by testing it internally, before publishing it as a Partner 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, Partner 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](#)
  - [Building CNF applications with OpenShift Pipelines](#)
- 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](#).

### 33.1.2. 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.

#### Task Summary

The certification workflow includes the following three primary stages-

1. [Section 33.1.2.1, “Certification onboarding for cnf”](#)
2. [Section 33.1.2.2, “Completing the product listing for cnf”](#)
3. [Section 31.4, “Publishing the product listing on the Red Hat Ecosystem Catalog”](#)

### 33.1.2.1. Certification onboarding for cnf

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. Create your product listing by selecting your desired product category. You can select from the available product categories:
  - a. Containerized Application
  - b. Standalone Application
  - c. OpenStack Infrastructure
4. Complete your company profile.



#### NOTE

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

#### Additional resources

For detailed instructions about creating your CNF product, see [Creating a product listing](#).

### 33.1.2.2. Completing the product listing for cnf

Perform the steps outlined for completing the checklist:

1. Provide details for your validation.
2. Validate the functionality of your CNF on Red Hat OpenShift for Partner Validation.
3. Complete the product listing details for certifying your CNF components.
4. Add components to the product listing.
5. Certify components for your product listing.

#### Additional resources

For more details about completing the product listing, see [Adding certification components](#).

### 33.1.2.3. Publishing the product listing on the Red Hat Ecosystem Catalog

The **Certified** or **Partner Validated** CNF component must be added 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 Partner Validated and Certified CNF products on the [Red Hat Ecosystem Catalog](#) with the respective labels.

### Additional resources

- For more details about publishing your CNF component, see [Publishing the product listing on the Red Hat Ecosystem Catalog](#).

## 33.2. CREATE A PRODUCT

The product listing provides marketing and technical information, showcasing your product's features and advantages to potential customers. It lays the foundation for adding all necessary components to your product for certification.

### Prerequisites

Ensure that 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 platform results in a substandard experience then you must resolve the issues before certification.

### Procedure

Red Hat recommends completing all optional fields in the listing tabs for a comprehensive product listing. More information helps mutual customers make informed choices.

Red Hat encourages collaboration with your product manager, marketing representative, or other product experts when entering information for your product listing.

Fields marked with an asterisk (\*) are mandatory.

### Procedure

1. Log in to the [Red Hat Partner Connect Portal](#).
2. Go to the Certified technology portal tab and click **Visit the portal**.
3. On the header bar, click **Product management**.
4. From the **Listing and certification** tab click **Manage products**.
5. From the **My Products** page, click **Create Product**.  
A **Create New Product** dialog opens.
6. Enter the **Product name**.

7. From the **What kind of product would you like to certify?** drop-down, select the required product category and click **Create product**. For example, select **Containerized Application** for creating a containerized product listing.

A new page with your Product name opens. It comprises the following tabs:

- [Section 33.2.1, "Overview for CNF"](#)
- [Section 33.2.2, "Product Information for CNF"](#)
- [Section 33.2.3, "Components for CNF"](#)
- [Section 33.2.4, "Support for CNF"](#)

Along with the following tabs, the page header provides the **Product Score** details. Product Score evaluates your product information and displays a score. It can be:

- Fair
- Good
- Excellent
- Best

8. Click **How do I improve my score?** to improve your product score.
9. After providing the product listing details, click **Save** before moving to the next section.

### 33.2.1. Overview for CNF

This tab consists of a series of tasks that you must complete to publish your product:

- [Section 33.2.1.1, "Complete product listing details for CNF"](#)
- [Section 33.2.1.2, "Complete company profile information for CNF"](#)
- [Section 33.2.1.3, "Accept legal agreements for CNF"](#)
- [Section 33.2.1.4, "Add at least one product component for CNF"](#)
- [Section 33.2.1.5, "Certify components for your listing for CNF"](#)

#### 33.2.1.1. Complete product listing details for CNF

1. To complete your product listing details, click **Start**.  
The **Product Information** tab opens.
2. Enter all the essential product details and click **Save**.

#### 33.2.1.2. Complete company profile information for CNF

1. To complete your company profile information, click **Start**. After entering all the details, click **Submit**.
2. To modify the existing details, click **Review**. The **Account Details** page opens.

3. Review and modify the Company profile information and click **Submit**.

### 33.2.1.3. Accept legal agreements for CNF

To publish your product image, agree to the terms regarding the distribution of partner container images.

1. To accept the legal agreements, click **Start**.
2. To preview or download the agreement, click **Review**.

The **Red Hat Partner Connect Container Appendix** document displays. Read the document to know the terms related to the distribution of container images.

### 33.2.1.4. Add at least one product component for CNF

1. Click **Start**. You are redirected to the **Components** tab.  
To add a new or existing product component, click **Add component**.
  2. For adding a new component,
    - a. In the **Component Name** text box, enter the component name. This name is not published and is only for internal use.
-  **NOTE**  
Red Hat recommends including the product version in the component name to aid easy identification of the newly created CNF component. For example, <CompanyName ProductName> 1.2 - OCP 4.12.2)
- b. For **What kind of OpenShift component are you creating?** select the component that you wish to certify. For example, for certifying your CNF, select **Cloud Native Function (CNF)**
  - c. Click **Create new component**
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
    - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
    - b. Click **Attach existing component**.



#### NOTE

Create individual CNF components for each partner product version and its corresponding Red Hat base version. Also create and attach certification components such as container images and operator bundle or Helm chart, which are required for the certification. You can create more than one CNF component for a product.

### 33.2.1.5. Certify components for your listing for CNF

1. To certify the components for your listing, click **Start**. If you have existing product components, you can view the list of **Attached Components** and their details:

- a. Name
  - b. Certification
  - c. Security
  - d. Type
  - e. Created
  - f. Click more options to archive or remove the components
2. Select the components for certification.

After completing all the above tasks you will see a green tick mark corresponding to all the options.

The Overview tab also provides the following information:

1. **Product contacts** - Provides Product marketing and Technical contact information.
  - a. Click **Add contacts to product** to provide the contact information
  - b. Click **Edit** to update the information.
2. **Components in product** - Provides the list of the components attached to the product along with their last updated information.
  - a. Click **Add components to product** to add new or existing components to your product.
  - b. Click **Edit components** to update the existing component information.

After publishing the product listing, you can view your **Product Readiness Score** and **Ways to raise your score** on the **Overview** tab.

### 33.2.2. Product Information for CNF

Through this tab you can provide all the essential information about your product. The product details are published along with your product on the Red Hat Ecosystem catalog.

**General** tab:

Provide 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. Click **Save**.

**Features & Benefits** tab:

Provide important features of your product.

1. Optional: Enter the **Title** and **Description**.
2. Optional: To add additional features for your product, click **+ Add new feature**.

3. Click **Save**.

#### Quick start & Config tab:

Add links to any quick start guide or configuration document to help customers deploy and start using your product.

1. Optional: Enter **Quick start & configuration instructions**
2. Click **Save**.
3. Select **Hide default instructions** check box, if you don't want to display them.

#### Linked resources tab:

Add links to supporting documentation to help our customers use your product. The information is mapped to and is displayed in the Documentation section on the product's catalog page.



#### NOTE

It is mandatory to add a minimum of three resources. Red Hat encourages you to add more resources, if available.

1. Select the **Type** drop-down menu, and enter the **Title** and **Description** of the resource.
2. Enter the **Resource URL**.
3. Optional: To add additional resources for your product, click **+ Add new Resource**.
4. Click **Save**.

#### FAQs tab:

Add frequently asked questions and answers of the product's purpose, operation, installation, or other attribute details. You can include common customer queries about your product and services.

1. Enter **Question** and **Answer**.
2. Optional: To add additional FAQs for your product, click **+ Add new FAQ**.
3. Click **Save**.

#### 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**.

#### Contacts tab:

Provide contact information of your marketing and technical team.

1. Enter the **Marketing contact email address** and **Technical contact email address**.

2. Optional: To add additional contacts, click + **Add another**.

3. Click **Save**.

#### Legal tab:

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:

Use this tab to improve the 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.

3. Click **Save**.

4. Optional: To add additional key-value pair, click + **Add new key-value pair**.



#### NOTE

Add at least one Search alias for your product. Red Hat encourages you to add more aliases, if available.

### 33.2.3. Components for CNF

Use this tab to add components to your product listing. Through this tab you can also view a list of attached components linked to your Product Listing.

Alternatively, to attach a component to the Product Listing, you can complete the **Add at least one product component** option available in the **Overview** tab of a Container, Operator, or Helm Chart product listing.

1. To add a new or existing product component, click **Add component**.

2. For adding a new component,

- a. In the **Component Name** text box, enter the component name.

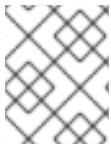
- b. In the **Component Name** text box, enter the component name. This name is not published and is only for internal use.



#### NOTE

Red Hat recommends including the product version in the component name to aid easy identification of the newly created CNF component. For example, **<CompanyName ProductName> 1.2 - OCP 4.12.2**

- c. For **What kind of OpenShift component are you creating?** select the component that you wish to certify. For example, for certifying your CNF, select **Cloud Native Function (CNF)**
  - d. Click **Create new component**
3. For adding an existing component, from the **Add Component** dialog, select **Existing Component**.
    - a. From the **Available components** list, search and select the components that you wish to certify and click the forward arrow. The selected components are added to the **Chosen components** list.
    - b. Click **Attach existing component**.



#### NOTE

You can add one component to multiple products listings. All attached components must be published before the product listing can be published.

After attaching components, you can view the list of **Attached Components** and their details:

- i. Name
- ii. Certification
- iii. Security
- iv. Type
- v. Created
- vi. Click more options to archive or remove the attached components

Alternatively, to search for specific components, type the component's name in the **Search by component Name** text box.

#### 33.2.3.1. Certify components for your listing

1. To certify the components for your listing, click **Start**. If you have existing product components, you can view the list of **Attached Components** and their details:
  - a. Name
  - b. Certification
  - c. Security
  - d. Type
  - e. Created
  - f. Click more options to archive or remove the components
2. Select the components for certification.

#### 33.2.4. Support for CNF

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

You can also contact the Red Hat Partner Acceleration Desk for any technical questions you may have regarding the Certification. Technical help requests will be redirected to the Certification Operations team.

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.

1. To request support, click Open a support case. See [PAD - How to open & manage PAD cases](#), to open a PAD ticket.
2. To view the list of existing support cases, click **View support cases**.

### 33.2.5. Removing a product

After creating a product listing if you wish to remove it, go to the **Overview** tab and click **Delete**.

A published product must first be unpublished before it can be deleted. Red Hat retains information related to deleted products even after you delete the product.

## 33.3. ADDING CERTIFICATION COMPONENTS

After creating the new product listing, add the certification components for the newly created product listing.

You can configure the following options for the newly added components:



### NOTE

The component configurations differ for different product categories.

- [Section 33.3.1, “Certification for CNF”](#)
- [Section 33.3.2, “Component details for CNF”](#)
- [Section 33.3.3, “Contact Information for CNF”](#)
- [Section 33.3.4, “Associated products for CNF”](#)

### 33.3.1. Certification for CNF

Validate and certify the functionality of your CNF on Red Hat OpenShift by using the **Certification** tab.

#### 33.3.1.1. Validate the functionality of your CNF on Red Hat OpenShift

By using this feature the Red Hat CNF certification team checks if your product meets all the standards for Partner Validation.

To validate the functionality of your CNF component, 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 modify the existing details, click **Review**. 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 **Partner 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.

### 33.3.1.2. Certify the functionality of your CNF on Red Hat OpenShift



#### NOTE

Select this option only if you want to certify your partner validated CNF component.

This is an optional feature that allows you to certify your Partner Validated component by using the Red Hat certification tool. For every Partner Validated component, 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 Partner Validated CNF component then it will be displayed on the [Red Hat Ecosystem Catalog](#) with the **Certified** label.

#### Prerequisites

1. Complete the product listing before proceeding with the certification.
2. Certify your attached container images, operator bundles or helm charts before submitting your CNF component for certification.

#### Procedure

To certify your Partner Validated CNF component, perform the following steps:

1. Go to the **Certification** tab and from the **Certify the functionality of your CNF on Red Hat OpenShift** tile, click **Start**. A new functional certification request is created and will be redirected to your component on the [Red Hat Partner Certification](#) (rhcrt) portal.
2. Run the [Red Hat Best Practices Test Suite for Kubernetes](#) 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 component, perform the following steps on your CNF component page on the [Red Hat Partner Certification](#) (rhcrt) portal:
  - a. Go to the **Summary** tab,

- i. To submit your product certification test results, from the **Files** section click **Upload**. Select the **claims.json** and **tnf\_config.yml** files. Then, click **Next**. You can see a successful upload message.
  - 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 component. 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 component. 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 component. 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 product must be recertified before beginning to certify a CNF component in the prescribed order.

After successful verification by the Red Hat CNF certification team, your Partner Validated CNF component will become certified, and will be automatically published on the [Red Hat Ecosystem Catalog](#) with the **Certified** label.

#### Additional resources

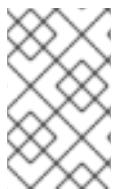
1. For more information about the Red Hat Best Practices Test Suite for Kubernetes, see the following resources:
  - a. [Overview](#)
  - b. [Test Suite documentation](#)
  - c. [Recorded demonstration](#)
  - d. [Test Catalog](#)
  - e. [Best Practices guide](#)
2. For more information about installing and configuring DCI OpenShift App Agent, see [DCI OpenShift App Agent](#).

### 33.3.2. Component details for CNF

Configure the product component details by using this tab.

Enter the required details in the following fields:

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

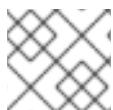


#### NOTE

Red Hat recommends including the product version to the component name to aid easy identification of the newly created CNF component. For example, **<CompanyName  
ProductName> 1.2 - OCP 4.12.2**)

- Click **Save**.

### 33.3.3. Contact Information for CNF



#### NOTE

Providing information for this tab is optional.

In the **Contact Information** tab, enter the primary technical contact details of your product component.

1. Optional: In the **Technical contact email address** field, enter the email address of the image maintainer.
2. Optional: To add additional contacts for your component, click **+ Add new contact**.
3. Click **Save**.

### 33.3.4. Associated products for CNF

The Associated Product tab provides the list of products that are associated with your product component along with the following information:

- Product Name

- Type - Traditional application
- Visibility - Published or Not Published
- Last Activity - number of days before you ran the test

To add products to your component, perform the following:

- If you want to find a product by its name, enter the product name in the **Search by name** text box and click the search icon.
- If you are not sure of the product name, click **Find a product**. From the **Add product** dialog, select the required product from the Available products list box and click the forward arrow. The selected product is added to the Chosen products list box. Click **Update attached products**. Added products are listed in the Associated product list.



#### NOTE

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

### 33.4. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit your CNF component for validation, the Red Hat CNF certification team will review and verify the entered details of the CNF Questionnaire. If you want to certify your Partner Validated CNF component, complete the Certification details.

The Red Hat certification team will review the submitted test results. After successful verification, to publish your product on the [Red Hat Ecosystem Catalog](#), go to the **Product Listings** page to attach the Partner Validated or Certified CNF component.

Follow these steps to publish your product listing:

1. Access the [Red Hat Partner connect](#) web page. **My Product** web page displays the **Product Listings**.
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. Go to the **Components** tab.
5. Click **Add Component** to add a new component to your product listing.



#### NOTE

You must publish all the attached components before publishing your product version and product certification.

6. Click **Attach Component** to attach your **Partner Validated or Certified** CNF component to this listing. While attaching a certified CNF component, it is mandatory to add the certified container image and an operator bundle or Helm chart used by your CNF component.

**NOTE**

All the attached components must be in **Published** status.

For Partner Validated components, this step is not required. The Publish button is enabled when you specify all the required information for the product listing, including the attached components.

#### 7. Click **Publish**.

Your new CNF product listing is now available for public access with respective **Partner 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, RHOCN 4 CNF
- Level - for example, Partner 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,

Product	Version	Partner product version	Certification type	Level
Red Hat OpenShift Container Platform	4.12	5.11	RHOCN 4 CNF	Partner Validated
Red Hat OpenShift Container Platform	4.12	5.12	RHOCN 4 CNF	Certified
Red Hat OpenShift Container Platform	4.12	5.13	RHOCN 4 CNF	Certified
Red Hat OpenShift Container Platform	4.12	5.14	RHOCN 4 CNF	Partner Validated

### 33.5. 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 component 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 all the tasks on the [overview](#) tab, 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](#).



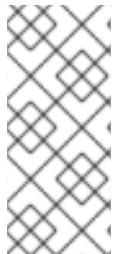
#### NOTE

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 34. CNI CERTIFICATION

### 34.1. WORKING WITH CONTAINER NETWORK INTERFACE (CNI) CERTIFICATION



#### NOTE

To certify a CNI plug-in, it must be configured to be deployed and managed by using an Operator. Before proceeding with the Red Hat CNI Certification, the operator, along with all the containers referenced by your CNI operator bundle, including the plug-in must already be certified and published on the [Red Hat Ecosystem Catalog](#) prior to certifying the CNI plug-in.

#### 34.1.1. 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.

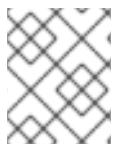
The CNI badge is a specialization within Red Hat OpenShift certification. It is available to products that implement a network function delivered in a container format by using a [CNI plug-in](#) with Red Hat OpenShift as the deployment platform.

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](#)

#### 34.1.2. 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.

#### Task Summary

The certification workflow includes three primary steps-

1. Section 34.1.2.1, "Certification on-boarding for CNI"
2. Section 34.1.2.2, "Certification testing for CNI"
3. Section 31.4, "Publishing the product listing on the Red Hat Ecosystem Catalog"

### 34.1.2.1. Certification on-boarding for CNI

Perform the steps outlined for certification onboarding:

1. Join the [Red Hat Connect](#) for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Create your product listing by selecting your desired product category. You can select from the available product categories:
  - a. Containerized Application
  - b. Standalone Application
  - c. OpenStack Infrastructure
4. Complete your company profile.
5. Add components to the product listing.
6. Certify components for your product listing.

#### Additional resources

For detailed instructions about creating a product listing, see [Creating a product listing](#).

### 34.1.2.2. Certification testing for CNI

Follow these high-level steps to run a 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 you face any issues.
4. Submit the certification results to Red Hat through a pull request.

#### Additional resources

For detailed instructions about attaching product components and validating the functionality of your CNI operators on Red Hat OpenShift, see [Adding certification components](#).

### 34.1.2.3. Publishing the 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.

## Additional resources

For detailed instructions about publishing the product listing on the Red Hat Ecosystem Catalog, see [Publishing the product listing on the Red Hat Ecosystem Catalog](#).

## 34.2. CREATING A PRODUCT

For detailed instructions about creating a product listing, see [Creating a product listing](#).

## 34.3. ADDING CERTIFICATION COMPONENTS

For detailed instructions about attaching product components and validating the functionality of your CNI operators on Red Hat OpenShift, see [Adding certification components](#).

## 34.4. 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](#).

## 34.5. 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 about 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 3 and the Istio resource to deploy an Istio control plane. For more information about migrating to Service Mesh 3, see [Migrating from Service Mesh 2 to Service Mesh 3](#).
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 Subscriptions](#).

## 34.6. 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:

```
$ mkdir -p test-results; cd test-results
$ podman run -v "$(pwd):/data:z" -w /data --rm -it $(KUBECONFIG=$(pwd)/kubeconfig.yaml
oc adm release info --image-for=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 exact version of the Red Hat OpenShift cluster, for example, 4.x.

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 --plugin-env kubevirt-
conformance.E2E_LABEL='(conformance && sig-network)'
```

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 [Sail Operator repository](#).
  - b. Read the detailed instructions provided in the [README.md](#) file in the repository explaining how to run the tests.
  - c. Set the appropriate flags corresponding to your OpenShift Service Mesh (OSSM) operator deployment, as outlined in the [README.md](#) file.
  - d. Run the tests and wait until they are completed.
  - e. The test suite generates a report.xml file. Submit this file and your CNI conformance test results, to the Red Hat certification team for verification.
4. To run the Red Hat OpenShift hosted control planes conformance test suite, place the **kubeconfig.yaml** file for the hosted cluster in the current working directory and run the following command:

```
$ mkdir -p test-results; cd test-results
$ podman run -v "$(pwd):/data:z" -w /data --rm -it $(KUBECONFIG=$(pwd)/kubeconfig.yaml
oc adm release info --image-for=tests) sh -c "KUBECONFIG=/data/kubeconfig.yaml
/usr/bin/openshift-tests run openshift/conformance/parallel -o /data/results.txt"
```

The command uses the test suite that corresponds to the exact version of the Red Hat OpenShift cluster, for example, 4.x.

## 34.7. 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.

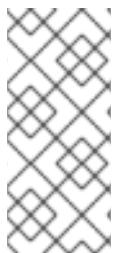
## 34.8. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit the CNI component for validation through the Red Hat certification portal , the Red Hat certification team reviews and verifies your component 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 Products** web page displays the **Product Listings**.
2. 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. Go to the **Components** tab.
5. Click **Add Component** to attach your certified CNI operator to this listing. Also add the certified containers used by your CNI component. All the components 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 components.
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 35. CSI CERTIFICATION

## 35.1. WORKING WITH CONTAINER STORAGE INTERFACE (CSI) CERTIFICATION



### NOTE

To certify a CSI plug-in, it must be configured to be deployed and managed by using an Operator. Before proceeding with the Red Hat CSI Certification, the operator, along with all the containers referenced by your CSI operator bundle, including the plug-in must already be certified and published on the [Red Hat Ecosystem Catalog](#) prior to certifying the CSI driver.

### 35.1.1. 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](#)

### 35.1.2. 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.

### Task Summary

The certification workflow includes three primary steps-

1. [Section 35.1.2.1, "Certification on-boarding for CSI"](#)
2. [Section 35.1.2.2, "Certification testing for CSI"](#)
3. [Section 31.4, "Publishing the product listing on the Red Hat Ecosystem Catalog"](#)

### 35.1.2.1. Certification on-boarding for CSI

Perform the steps outlined for certification onboarding:

1. Join the [Red Hat Connect](#) for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Create your product listing by selecting your desired product category. You can select from the available product categories:
  - a. Containerized Application
  - b. Standalone Application
  - c. OpenStack Infrastructure
4. Complete your company profile.
5. Add components to the product listing.
6. Certify components for your product listing.

#### Additional resources

For detailed instructions about creating a product listing, see [Creating a product listing](#).

### 35.1.2.2. Certification testing for CSI

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.

#### Additional resources

For detailed instructions about attaching product components and validating the functionality of your CNI operators on Red Hat OpenShift, see [Adding certification components](#).

### 35.1.2.3. Publishing the 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.

#### Additional resources

For detailed instructions about publishing the product listing on the Red Hat Ecosystem Catalog, see [Publishing the product listing on the Red Hat Ecosystem Catalog](#).

## 35.2. CREATING A PRODUCT

For detailed instructions about creating a product listing, see [Creating a product listing](#).

## 35.3. ADDING CERTIFICATION COMPONENTS

For detailed instructions about attaching product components and validating the functionality of your CNI operators on Red Hat OpenShift, see [Adding certification components](#).

## 35.4. 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](#)
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](#).

## 35.5. 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, such as OpenShift 4.17 or later. Red Hat recommends using the latest supported release. For more information about 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.

[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 Subscriptions](#).

## 35.6. 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.

Access the catalog entries for OpenShift End-to-End tests container at the following locations:

- [Red Hat OpenShift v4.15 and earlier](#)
- [Red Hat OpenShift v4.16 onwards](#)



### NOTE

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.

## 35.7. 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](#).

## 35.8. RUNNING THE CSI CERTIFICATION TESTS

The CSI certification test comprises two types of tests:

- [CSI function test](#)
- [KubeVirt storage checkup test](#)

Run both the tests to certify your products for CSI certification.

### 35.8.1. Running the CSI function test

On the test client, place the **kubeconfig.yaml** and **manifest.yaml** files in the current working directory and run the following command:

**For OpenShift v4.16 and later:**

```
$ mkdir -p test-results; cd test-results
$ VERSION=$(oc get clusterverSION -o 'jsonpath={.items[0].status.desired.version}' | cut -d. -f1,2)
$ podman run -v "$pwd:/data:z" -w /data --rm -it registry.redhat.io/openshift4/ose-tests-rhel9:${VERSION} sh -c "KUBECONFIG=/data/kubeconfig.yaml
TEST_CSI_DRIVER_FILES=/data/manifest.yaml /usr/bin/openshift-tests run openshift/csi -o /data/results.txt"
```

### For OpenShift v4.15 and earlier:

```
$ mkdir -p test-results; cd test-results
$ VERSION=$(oc get clusterversion -o 'jsonpath={.items[0].status.desired.version}' | cut -d. -f1,2)
$ podman run -v "$pwd":/data:z -w /data --rm -it registry.redhat.io/openshift4/ose-tests:${VERSION}
sh -c "KUBECONFIG=/data/kubeconfig.yaml TEST_CSIDRIVER_FILES=/data/manifest.yaml
/usr/bin/openshift-tests run openshift/csi -o /data/results.txt"
```

The command uses the test suite that corresponds to the exact version of the Red Hat OpenShift cluster, for example, 4.x.

If you want to see a list of the tests that are run for CSI certification, run the following command:

### For OpenShift v4.16 and later:

```
podman run -v `pwd`:/data:z --rm -it registry.redhat.io/openshift4/ose-tests-rhel9 sh -c
"KUBECONFIG=/data/kubeconfig.yaml TEST_CSIDRIVER_FILES=/data/manifest.yaml
/usr/bin/openshift-tests run --dry-run openshift/csi"
```

### For OpenShift v4.15 and earlier:

```
podman run -v `pwd`:/data:z --rm -it registry.redhat.io/openshift4/ose-tests sh -c
"KUBECONFIG=/data/kubeconfig.yaml TEST_CSIDRIVER_FILES=/data/manifest.yaml
/usr/bin/openshift-tests run --dry-run openshift/csi"
```



#### NOTE

Execute separate tests for each supported storage protocol.

### Saving the CSI test results

Make sure to have the following ready before submitting your test results:

- Contents on 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
```

### 35.8.2. Running the KubeVirt storage checkup test

The KubeVirt storage checkup test is an automated tool that verifies if the cluster storage is optimally configured for OpenShift Virtualization. You can run the test by using two different methods – web console or CLI.

#### 35.8.2.1. Running a Kubevirt storage checkup test by using the web console

Run a storage checkup test by using the web console to validate if the configured storage is optimally configured for virtual machines. For detailed instructions, see [Running a storage checkup by using the web console](#).

### 35.8.2.2. Running a KubeVirt storage checkup test by using the command line

Run a storage checkup test by using the command line to validate if the configured storage is optimally configured for virtual machines. For detailed instructions, see [Running a storage checkup by using the command line](#).

#### Saving the KubeVirt storage checkup test results

Make sure to have the output of the following command ready before submitting your test results:

```
$ oc get configmap storage-checkup-config -n <target_namespace> -o yaml
```

#### Additional resources

For more information about the KubeVirt storage checkup test, see [kubevirt-storage-checkup](#).

## 35.9. SUBMITTING CSI TEST RESULTS

Follow the procedure to submit both 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.



#### NOTE

If any modifications are made to the CSI driver while running the KubeVirt storage test or the CSI function test, you must rerun both the CSI certification tests.

## 35.10. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit the CSI component for validation through the Red Hat certification portal, the Red Hat certification team reviews and verifies your component 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 Products** web page displays the **Product Listings**.
2. 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. Go to the **Components** tab.
5. Click **Add Component** to attach your certified CSI operator to this listing. Also add any additional certified containers used by your CSI component. All the components 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 components.
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.