# Table of Contents

* [Table of Contents](#table-of-contents)
* [Automate certification process by utilizing DCI to interact with the certification portal backend](#X56606fd4e63c68028fdb1c4a5ebefc59a37ff79)
  + [Prerequisites](#prerequisites)
  + [Automation Container Certification Flow](#automation-container-certification-flow)
    - [Auto Publish Preparations](#auto-publish-preparations)
    - [Auto Publish Settings Configuration](#auto-publish-settings-configuration)
  + [Recertify Certification Container Projects](#Xbbc18839561a9bdaa8f13bc36c037b1fabc17d0)
    - [Recertify Settings Configuration](#recertify-settings-configuration)
  + [Automate creation of the Openshift-cnf project for Vendor Validated](#Xf05d4ba2f4d82fdcf29552e875db254eba587a3)
    - [Global Variables](#global-variables)
    - [Variables to define for each cnf\_to\_certify](#X9a3f3b35b3a3fdeb37bad81466f6b4e8b9f7529)
    - [Variables to define for project settings under cert\_listings main variable](#Xe416ea13a4346136e36a2674ae5815e7da8592f)
    - [Example Configuration of Openshift-cnf certification project creation](#X2f53adb0251ec03bf9c948ee9ab047701149adf)
  + [Run Chart Verifier and TNF Suite Test Together](#X65f4ad30139de861551328f92a07f2502f6aa37)
    - [Example of chart-verifier and TNF Test Suite Settings Configuration](#X9e76b3cc4cceb58169d9d8f94d2bf5bbec770eb)
  + [How to Run DCI Auto-publish, Recertify and Openshift-cnf Vendor validated](#Xdcf62536e5050a51def516ce7ff28a985e06e29)
  + [Known Issues](#known-issues)
  + [Links](#links)

# Automate certification process by utilizing DCI to interact with the certification portal backend

This repository provides an automated certification process for container projects, along with their automatic publication. It encompasses both new container certification and container new release recertification by leveraging the new features offered by DCI. It also includes new procedure of Openshift-cnf certification project auto creation for Vendor Validate and CNF certification.

The process for certifying a new container involves creating container projects, adding or modifying mandatory project parameters, testing and scanning container images, attaching to a product listing, and auto publishing container projects once all requirements are fulfilled.

The process for recertifying existing container projects involves retesting and rescanning new release container images, updating the container’s version, its digest and tag, and republishing to the catalog.

Finally, an integrated process for creating the Openshift-cnf certification project has been introduced into the workflow. Previously, the partner CNF Vendor Validate required a separate steps. However, with the new enhancement implemented in DCI, once the container and helm chart projects are certified, the Openshift-cnf project will be automatically created.

## Prerequisites

* Prepare a jumphost to install and run DCI then following this [link](https://doc.distributed-ci.io/dci-openshift-agent/#installation-of-dci-jumpbox)
* Minimum DCI Openshift APP Agent version (Recommended to use latest DCI software version)
* Upgrade or re-install the latest DCI Repo Follow this link if upgrade/remove/install NOT workking [install-dci-packages](https://blog.distributed-ci.io/install-openshift-on-baremetal-using-dci.html#dci-packages)

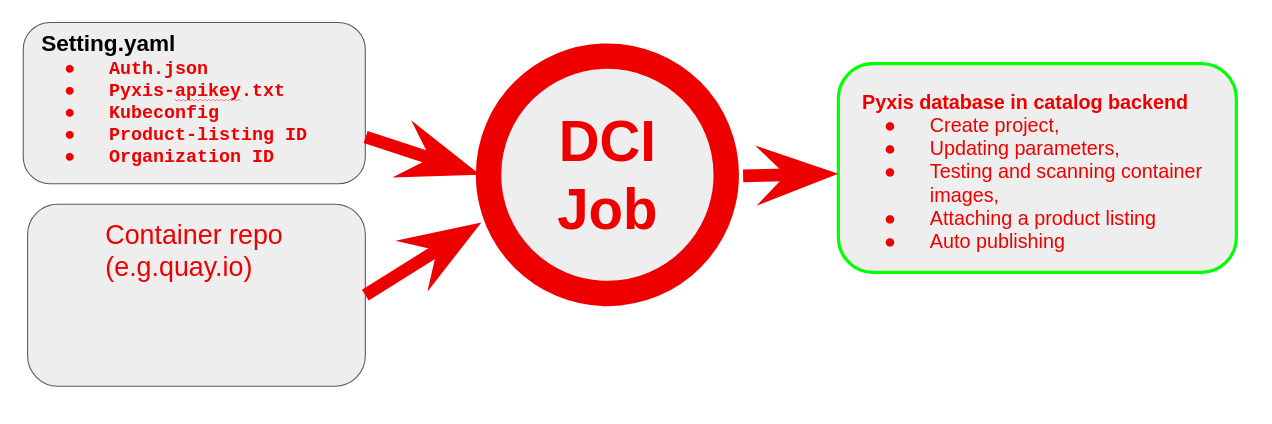
$ sudo dnf upgrade --refresh --repo dci -y

* if upgrade gets issue then try following:

$ sudo dnf remove dci-openshift-app-agent -y  
$ sudo dnf install dci-openshift-app-agent -y

* DCI Control server credential [create remote-ci credentials](https://www.distributed-ci.io/remotecis)
* Prepare settings.yml for container and CNF projects information  
  The details of each container certification project type are shown on next sections
* Set auto-publish parameter to on under container project settings tab

## Automation Container Certification Flow

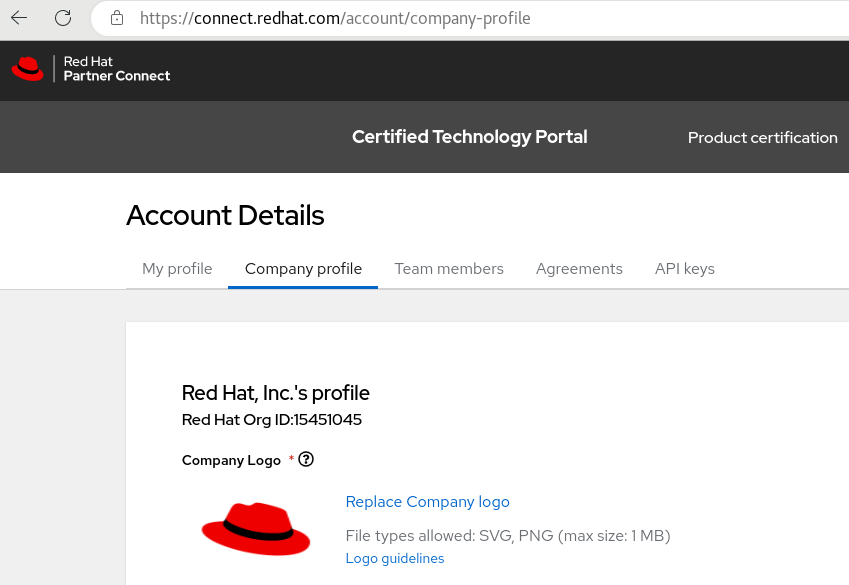


Automation Container Cert Workflow

### Auto Publish Preparations

* OCP and DCI ENV (included dcirc.sh,install.yml etc..)  
  Any OCP Cluster and a helper node with DCI RPM packages installed
* **Settings.yml**  
  A Settings with DCI auto-publish enhancement feature
* **Auth.json**  
  Docker Authentication to access container repo on private registry server  
  Login to your private registry server:

$ podman login -u testuser quay.io  
$ echo $XDG\_RUNTIME\_DIR  
/run/user/4205315/containers/auth.json

* **Pyxis-apikey.txt**  
  A token to access specific partner Pyxis catalog data using REST API. [Create Pyxis API Key](https://connect.redhat.com/account/api-keys)
* **Kubeconfig**  
  A kubeconfig that can access the OCP cluster
* **Product-listing ID**  
  Before a container or helm chart/operator can be publicly listed into RedHat catalog, a Product-Listing must be created, it only need to create once according to CNF type. Follow this link to [Create-Product-Listing](https://connect.redhat.com/manage/products)
* **Organization ID**  
  Mandatory when using create\_container\_project. Company ID will be used for the verification of container certification project Organization-ID Company-Profile. 

### Auto Publish Settings Configuration

---  
dci\_topic: OCP-4.11  
dci\_name: Testing DCI to create certification Project Automatic and Update mandatory settings and publish  
dci\_configuration: Using DCI create project,update,submit result and auto-publish  
preflight\_test\_certified\_image: true  
check\_for\_existing\_projects: true  
ignore\_project\_creation\_errors: true  
dci\_config\_dirs: [/etc/dci-openshift-agent]  
partner\_creds: "/var/lib/dci-openshift-app-agent/auth.json"  
organization\_id: 12345678  
preflight\_containers\_to\_certify:  
 - container\_image: "quay.io/avu0/auto-publish-ubi8-nginx-demo1:v120"  
 create\_container\_project: true  
 short\_description: "I am doing a full-automation e2e auto-publish for following image auto-publish-ubi8-nginx-demo1:v120"  
 attach\_product\_listing: true  
  
 - container\_image: "quay.io/avu0/auto-publish-ubi8-nginx-demo2:v120"  
 create\_container\_project: true  
 short\_description: "I am doing a full-automation e2e auto-publish for following image auto-publish-ubi8-nginx-demo2:v120"  
 attach\_product\_listing: true  
  
cert\_settings:  
 auto\_publish: true  
 build\_categories: "Standalone image"  
 registry\_override\_instruct: "<p>This is an instruction how to get the image link.</p>"  
 email\_address: "whoami@redhat.com"  
 application\_categories: "Networking"  
 os\_content\_type: "Red Hat Universal Base Image (UBI)"  
 privileged: false  
 release\_category: "Generally Available"  
 repository\_description: "This is a test for Demo how to automate to create project, SCAN and update settings"  
  
cert\_listings:  
 published: true  
 type: "container stack"  
 pyxis\_product\_list\_identifier: "yyyyyyyyyyyyyyyyy" # product list id for container projects  
  
pyxis\_apikey\_path: "/var/lib/dci-openshift-app-agent/pyxis-apikey.txt"  
dci\_gits\_to\_components: []  
...

## Recertify Certification Container Projects

In the auto-publish process, we have seen the effectiveness of using DCI to communicate with the catalog backend, which enables full end-to-end automation for container certification.

In the next recertification process, we can use a similar approach to recertify a new release of a certified container under an existing project. DCI is also able to recertify certification projects that are already active and published with a new release version and tag.

* Identify the certification projects that need to be recertified
* Update the certification project’s release version and tag to the new release version and tag
* Run DCI to embed Preflight to retest the container of new release and submit new report to current container project in partner portal
* Confirm and verify the project information under the project, then publish it to the catalog for completion recertification.

Please note that these are just general steps and may vary depending on the specific project. And also basic and general requirements to run recertify the container projects please use follow this [section](https://github.com/ansvu/cnf-certification-automation-with-dci/tree/main#auto-publish-preparations)

### Recertify Settings Configuration

---  
dci\_topic: OCP-4.11  
dci\_name: Testing DCI to Recertify the certification container projects  
dci\_configuration: Using DCI to Recertify the Certification container Project  
preflight\_test\_certified\_image: true  
check\_for\_existing\_projects: true  
ignore\_project\_creation\_errors: true  
organization\_id: 12345678  
dci\_config\_dirs: [/etc/dci-openshift-agent]  
partner\_creds: "/var/lib/dci-openshift-app-agent/auth.json"  
preflight\_containers\_to\_certify:  
 - container\_image: "quay.io/avu0/auto-publish-ubi8-nginx-demo1:v121"  
 create\_container\_project: true  
   
 - container\_image: "quay.io/avu0/auto-publish-ubi8-nginx-demo2:v121"  
 create\_container\_project: true  
  
pyxis\_apikey\_path: "/var/lib/dci-openshift-app-agent/pyxis-apikey.txt"  
dci\_gits\_to\_components: []  
...

As you may have noticed, the create\_container\_project parameter is now set to true even when recertifying container projects. This is a new change that was recently made. The backend will now reject the creation of a container project if it is already in the active state, but the DCI will ignore this and continue with the recertification process.

## Automate creation of the Openshift-cnf project for Vendor Validated

This feature automatically generates an OpenShift-CNF certification project when this option create\_cnf\_project set to true. The new feature reuses some tasks from the create-certification-project role, and the associated templates are stored within this feature. Currently, there are no mandatory parameters that need to be updated in the new feature.

Please note that the feature openshift-cnf is currently a basic automation setup. It will undergo further updates once additional options are available from the backend REST API, such as {"certification\_level":"Certified"} parameter for starting the Certify the functionality of your CNF on Red Hat OpenShift step within the project.

### Global Variables

As the new role openshift-cnf reuses some existing tasks, please refer to the description in the create-certification-project role for information on the shared global variables.

### Variables to define for each cnf\_to\_certify

| Name | Default | Description |
| --- | --- | --- |
| attach\_product\_listing | false | If set to true, it would attach product-listing to Openshift-cnf certification project. |
| create\_cnf\_project | false | If set to true, it would create a new Openshift-cnf certification project. |
| cnf\_name | None | If defined, it would create Openshift-cnf certification project for vendor validated, cnf\_name format: CNF25.8 + OCP4.12 |

### Variables to define for project settings under cert\_listings main variable

| Name | Default | Description |
| --- | --- | --- |
| pyxis\_product\_list\_identifier | None | Product-listing ID, it has to be created before. [See doc](https://redhat-connect.gitbook.io/red-hat-partner-connect-general-guide/managing-your-account/product-listing) |
| published | false | Boolean to enable publishing list of products |
| type | “container stack” | String. Type of product list |
| email\_address | “mail@example.com” | String. Email address is needed for creating openshift-cnf project |

### Example Configuration of Openshift-cnf certification project creation

---  
dci\_topic: OCP-4.11  
dci\_name: Testing Openshift-cnf auto creation and attach  
dci\_configuration: Using DCI create cnf project and attach product-list  
check\_for\_existing\_projects: true  
ignore\_project\_creation\_errors: true  
dci\_config\_dirs: [/etc/dci-openshift-agent]  
partner\_creds: "/var/lib/dci-openshift-app-agent/auth.json"  
organization\_id: 12345678  
#cnf\_name is a free-text but format: CNF-version + OCP-version e.g "CNF23.5 OCP4.12.9"  
cnf\_to\_certify:  
 - cnf\_name: "test-smf23.5 OCP4.11.5"  
 create\_cnf\_project: true  
 attach\_product\_listing: true  
  
 - cnf\_name: "test-upf23.5 OCP4.11.5"  
 create\_cnf\_project: true  
 attach\_product\_listing: true  
  
cert\_listings:  
 #email\_address is mandatory when creating openshift-cnf for vendor validation but does not hurt to define it  
 email\_address: "email@example.com"  
 published: false  
 type: "container stack"  
 pyxis\_product\_list\_identifier: "yyyyyyyyyyyyyyyyy" #7GC UDM  
  
pyxis\_apikey\_path: "/var/lib/dci-openshift-app-agent/pyxis-apikey.txt"  
dci\_gits\_to\_components: []  
...

**Note:** New and recertified container image projects can be included in the same settings.yml file, but new container certification projects require more detailed descriptions and additional parameters in the cert\_setting section. This is so that partners can easily distinguish between new and recertified images.

For recertified container projects, if partners have not yet enabled auto-publish for the projects from the portal Gui, they must manually enable it before using the DCI to automate the certification process.

## Run Chart Verifier and TNF Suite Test Together

This section will show to use DCI to run chart-verifier with option -c or --skip-cleanup so CNF helm chart deploys and leave CNF PODs running then using DCI to test the TNF suite base CNF namespace.

### Example of chart-verifier and TNF Test Suite Settings Configuration

---  
dci\_topic: OCP-4.11  
dci\_name: Chart-verifier with TNF Suite testing  
dci\_configuration: DCI Chart-verifier SampleChart + TNF Suite testing  
dci\_openshift\_app\_image: quay.io/testnetworkfunction/cnf-test-partner:latest  
do\_chart\_verifier: true  
dci\_openshift\_app\_ns: avachart  
dci\_teardown\_on\_success: false  
do\_must\_gather: false  
check\_workload\_api: false  
dci\_disconnected: false  
partner\_name: telcoci SampleChart  
partner\_email: telco.sample@redhat.com  
github\_token\_path: "/opt/cache/token.txt"  
dci\_charts:  
 - name: samplechart2  
 chart\_file: https://github.com/ansvu/samplechart2/releases/download/samplechart-0.1.3/samplechart-0.1.3.tgz  
 #chart\_values: https://github.com/ansvu/samplechart2/releases/download/samplechart-0.1.3/values.yaml  
 #install: true  
 deploy\_chart: true  
 flags: "-c -W"  
 create\_pr: false  
  
do\_cnf\_cert: true  
tnf\_labels: common,telco,extended  
tnf\_log\_level: trace  
tnf\_config:  
 - namespace: avachart  
 targetNameSpaces:  
 - avachart  
 operators\_regexp:  
 exclude\_connectivity\_regexp:  
test\_network\_function\_version: v4.2.4  
dci\_gits\_to\_components: []  
...

Note: Some cases where partner wants to deploy CNF and leave it running and do not want to teardown. So with new feature from chart-verifier 1.12.1, it adds -c option to allow users to skip the cleanup e.g. helm uninstall.

Result can be seen from DCI Job Server then click [here](https://www.distributed-ci.io/jobs/c65dae62-d2bb-4b28-becf-ff0975130851)

## How to Run DCI Auto-publish, Recertify and Openshift-cnf Vendor validated

* Login to DCI user

$ su - dci-openshift-app-agent

* Prepare a settings file for different type of certification projects accordingly
  + [Auto Publish New Container Certification Project Settings](https://github.com/ansvu/cnf-certification-automation-with-dci/tree/main#auto-publish-settings-configuration)
  + [Recertify Container Certification Project Settings](https://github.com/ansvu/cnf-certification-automation-with-dci/tree/main#recertify-settings-configuration)
  + [Openshift-cnf Certification Project Vendor Validated Settings](https://github.com/ansvu/cnf-certification-automation-with-dci/tree/main#example-configuration-of-openshift-cnf-certification-project-creation)
* Export KUBECONFIG

$ export KUBECONFIG=/var/lib/dci-openshift-app-agent/kubeconfig

* Start RUN DCI OpenShift App Agent

$ dci-openshift-app-agent-ctl -s -- -vv

## Known Issues

* Must gather log disable

For partners with larger OpenShift clusters and disconnected labs, running DCI for container image automation and recertification can pose challenges. Uploading the must-gather log, which collects OCP logging data, becomes problematic due to its size. This can cause DCI to get stuck for over 2 hours as it retries unsuccessfully through the partner’s proxy.

A solution is to disable must-gather log collection, the set do\_must\_gather: false to settings.yml. This allows DCI to proceed without uploading the large logs and avoids proxy-related issues.

## Links

* [dci-openshift-app-agent](https://doc.distributed-ci.io/dci-openshift-app-agent/)
* [dci-packages](https://blog.distributed-ci.io/install-openshift-on-baremetal-using-dci.html#dci-packages)