

Oracle Cloud Native Environment

Applications for Release 2



F96198-13
August 2025



Oracle Cloud Native Environment Applications for Release 2,

F96198-13

Copyright © 2024, 2025, Oracle and/or its affiliates.

Documentation License

The content in this document is licensed under the [Creative Commons Attribution–Share Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/) (CC-BY-SA) license. In accordance with CC-BY-SA, if you distribute this content or an adaptation of it, you must provide attribution to Oracle and retain the original copyright notices.

Contents

Preface

1 Application Catalogs

Oracle Catalog	1
External Catalogs	2

2 Managing Catalogs

Listing Catalogs	1
Getting a Catalog	1
Adding a Catalog	2
Removing a Catalog	2
Searching a Catalog	2
Updating the Oracle Catalog	3

3 Catalog Mirrors

Creating a Catalog Mirror	2
---------------------------	---

4 Managing Applications

Listing Installed Applications	1
Showing Application Details	2
Installing an Application	3
Updating an Application	4
Creating an Application Template	5
Installing an Application from a Template	6
Uninstalling an Application	7

Preface

This book discusses installing and managing applications and application catalogs available in Oracle Cloud Native Environment (Oracle CNE). This book provides information on using both the Oracle CNE Command Line Interface (CLI) (`ocne` command) and User Interface (UI) to work with applications and catalogs.

1

Application Catalogs

Learn about application catalogs and cloud native applications in Oracle Cloud Native Environment (Oracle CNE).

An application catalog is a searchable collection of software that can be installed into a Kubernetes cluster. Installed catalogs can be searched using both the Oracle CNE Command Line Interface (CLI) and the User Interface (UI).

Catalogs have a straightforward life cycle. They can be added and removed, but not changed.

Two types of application catalogs can be configured within a cluster: an Oracle catalog, and an external community catalog.

An application catalog is set up in two flavors: a Helm repository, and a service that's compatible with Artifact Hub (an external catalog). The Oracle catalog is a Helm repository, while an external catalog typically points to artifacthub.io and is compatible with Artifact Hub.

A Helm-based catalog is a collection of Helm artifacts, namely an `index.yaml` file, and a set of tarballs. These can be made available and served by any URI that can be read by an Oracle CNE component (the `ocne` CLI, the UI, and Helm).

An external catalog must be compatible with the Artifact Hub API. In practice, this means the external catalogs are served by an instance of Artifact Hub.

Oracle Catalog

The Oracle catalog is a collection of cloud native application software provided by Oracle. Oracle CNE applications are delivered through the Oracle catalog. The Oracle catalog can be accessed in two ways, either using the embedded version in the CLI, or from an application deployed to a Kubernetes cluster.

The embedded catalog is built into the CLI, and is named `embedded`. This catalog can be accessed and queried without deploying a Kubernetes cluster.

The Oracle catalog can also be deployed to a cluster, and is named `Oracle Cloud Native Environment Application Catalog`. The CLI deploys the `ocne-catalog` application from the Oracle Container Registry to a cluster in the `ocne-system` namespace. The `ocne-catalog` application includes a Helm repository, and an instance of NGINX that serves the static content. While the Oracle catalog can be served anywhere that has a container runtime, it's primarily intended to be run within the Kubernetes cluster that consumes its contents.

Note

The list of applications included in the Oracle catalog is available in the [upstream Oracle CNE documentation](#). To see the application list that matches the Oracle CNE release, select the release branch in the **Branches** drop down. For example:

- The branch name for Release 2.0 is [release/2.0](#).
- The branch name for Release 2.1 is [release/2.1](#).
- The branch name for Release 2.2 is [release/2.2](#).

The embedded Oracle catalog is updated when you update the CLI. The Oracle catalog running as an application in the cluster can be updated using the `ocne application update` command. This means that the catalog contents might differ. For more information on updating the Oracle catalog running in the cluster, see [Updating the Oracle Catalog](#).

External Catalogs

External catalogs can be added using the `ocne catalog add` CLI command. External catalogs are added using a Kubernetes external service resource. An external catalog isn't added to the cluster, instead, it's referenced. The catalog data isn't installed, or served from the cluster. An example of an external catalog is the Artifact Hub catalog. Artifact Hub is a web-based application that provides cloud native packages you can install into a Kubernetes cluster. For more information on Artifact Hub, see:

<https://artifacthub.io/>

2

Managing Catalogs

Learn how to work with application catalogs in Oracle CNE.

Steps are provided to use the Oracle CNE CLI (the `ocne` command) and the UI to perform actions. If the option to use one of these interfaces isn't provided, the option isn't available in that interface.

For more information on the CLI, see [Oracle Cloud Native Environment: CLI](#). For more information on setting up the UI, see [Oracle Cloud Native Environment: Kubernetes Clusters](#).

You can perform the following tasks to manage application catalogs:

- [Listing Catalogs](#)
- [Getting a Catalog](#)
- [Adding a Catalog](#)
- [Removing a Catalog](#)
- [Searching a Catalog](#)
- [Updating the Oracle Catalog](#)

Listing Catalogs

Use the `ocne catalog list` command to display the installed application catalogs. The syntax is:

```
ocne catalog {list|ls}
```

Getting a Catalog

Use the `ocne catalog get` command to list details about an installed application catalog. The syntax is:

```
ocne catalog get  
{-N|--name} name
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To display information about an application catalog:

```
ocne catalog get --name artifacthub
```

Adding a Catalog

Use the `ocne catalog add` command to add an application catalog. The syntax is:

```
ocne catalog add
{-N|--name} name
[{-n|--namespace} namespace]
[{-p|--protocol} protocol]
{-u|--uri} URI
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To add the Artifact Hub application catalog:

```
ocne catalog add --protocol artifacthub --name artifacthub --uri https://
artifacthub.io
```

Removing a Catalog

Removing a catalog doesn't uninstall applications from that catalog. Applications installed from a catalog that has subsequently been removed remain installed and must be uninstalled manually.

Use the `ocne catalog remove` command to remove an application catalog. The syntax is:

```
ocne catalog remove
{-N|--name} name
[{-n|--namespace} namespace]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To remove an application catalog named `artifacthub`:

```
ocne catalog remove --name artifacthub
```

Searching a Catalog

Search a catalog to find the list of applications that can be installed or updated.

-
- [CLI](#)
 - [UI](#)

CLI

Use the `ocne catalog search` command to search for applications in an application catalog. The syntax is:

```
ocne catalog search
[{-N|--name} name]
[{-p|--pattern} pattern]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To show all applications in the Oracle catalog:

```
ocne catalog search
```

To show all applications in the embedded Oracle catalog, built into the CLI:

```
ocne catalog search --name embedded
```

To search the default catalog for a specific string:

```
ocne catalog search --pattern 'ingress-*
```

UI

To search for applications in the default catalog:

1. In the navigation menu, select **Apps**.
The **Applications** page is displayed, listing all the applications available in the default catalog.
2. Enter characters in the **Search** field to display only applications whose name contains the entered characters.

Updating the Oracle Catalog

The embedded Oracle catalog is updated when you update the CLI. The Oracle catalog running as an application in the cluster can be updated using the `ocne application update` command. This pulls the latest container image for the Oracle catalog and updates the application.

Use the `ocne application update` command to update the Oracle catalog. The syntax is:

```
ocne application update
{-b|--built-in-catalog}
[{-c|--catalog} name]
[{-n|--namespace} namespace]
{-r|--release} name
[--reset-values]
```

```
[{-u|--values} URI]  
[{-v|--version} version]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Example:

To update the Oracle catalog application running in the cluster:

```
ocne application update --built-in-catalog
```

3

Catalog Mirrors

Learn how to mirror an application catalog to a private container registry, using the CLI.

A typical Kubernetes deployment requires many container images to function. When these Kubernetes deployments don't have Internet access, such as those in air-gapped environments, it's time-consuming to move these images from the public container registry to another location that's accessible to the cluster nodes. The `ocne catalog mirror` command simplifies this process, by enabling users to migrate images used by applications in application catalogs between container registries.

Mirroring an application catalog makes a copy of the container images in an application catalog and copies them to another container registry. You can also create an archive file of the images and save them to the local disk.

The `ocne catalog mirror` command can clone either all the images that are available in an application catalog, or a subset of images based on a list of applications and their configuration in a cluster configuration file. For example, the following cluster configuration file pulls specific application images and their configuration settings from the default catalog, which uses the Oracle Container Registry:

```
applications:
  - name: ocne-catalog
  - name: ui
```

For information on the options available to use in this configuration file, see the `applications` option in a cluster configuration file in [Oracle Cloud Native Environment: Kubernetes Clusters](#).

The host that runs the CLI must have access to all the relevant container registries, using the Internet, internal networks, or a combination. A private container registry must be on a host that's accessible to all nodes within the cluster.

You can use an existing container registry on the network, or create one for this specific purpose. Oracle provides tools which can be used to create container registries. To use Podman to create a private container registry, see the [Oracle Linux: Podman User's Guide](#).

To pull the images from the default catalog, access to an existing cluster must be set using the `kubeconfig` file. Creating an ephemeral cluster is an easy method to get access to the default catalog. To start an ephemeral cluster use:

```
ocne cluster start
export KUBECONFIG=$(ocne cluster show --cluster-name ocne)
```

If the default catalog isn't deployed to a cluster, or isn't available, the embedded catalog (built into the CLI) is used to mirror the container images.

To see a list of the images that can be mirrored to a private registry, use:

```
ocne catalog mirror
```

Some images aren't included when mirroring a catalog, such as `flannel` and `kube-proxy`. The excluded images are part of the OCK image, so aren't needed in a catalog mirror.

Creating a Catalog Mirror

Mirror the images in an application catalog to a private container registry.

Images are only pushed to a private container registry when you specify the `--push` option.

Use the `ocne catalog mirror` command to mirror the container images to a local container registry. The syntax to use is:

```
ocne catalog mirror
[{-a|--archive} path]
[{-c|--config} path]
{-d|--destination} URI
[{-o|--download}]
[{-N|--name} name]
[{-p|--push}]
[{-s|--source} registry]
[{-q|--quiet}]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Example 3-1 Mirror the Oracle catalog

To mirror the Oracle catalog to a private container registry:

```
ocne catalog mirror --destination myregistry.example.io --push
```

Example 3-2 Mirror the embedded Oracle catalog

To mirror the Oracle catalog embedded in the CLI to a private container registry:

```
ocne catalog mirror --name embedded --destination myregistry.example.io --push
```

Example 3-3 Mirror specific applications

To mirror only those images that are used by the applications listed in a cluster configuration file to a private container registry:

```
ocne catalog mirror --destination myregistry.example.io --config
mycluster.yaml --push
```

Example 3-4 Download specific applications to a named archive file

To download the images listed in a cluster configuration file to a specified local archive file:

```
ocne catalog mirror --config mycluster.yaml --download --archive $HOME/
myimages.tgz
```

4

Managing Applications

Learn how to work with applications in Oracle CNE.

Steps are provided to use the Oracle CNE CLI (the `ocne` command) and the UI to perform actions. If the option to use one of these interfaces isn't provided, the option isn't available in that interface.

For more information on the CLI, see [Oracle Cloud Native Environment: CLI](#). For more information setting up the UI, see [Oracle Cloud Native Environment: Kubernetes Clusters](#).

You can perform the following tasks to manage applications:

- [Listing Installed Applications](#)
- [Showing Application Details](#)
- [Installing an Application](#)
- [Updating an Application](#)
- [Creating an Application Template](#)
- [Installing an Application from a Template](#)
- [Uninstalling an Application](#)

Listing Installed Applications

-
- CLI
 - UI

CLI

Use the `ocne application list` command to list the installed applications. The syntax is:

```
ocne application {list|ls}
[{-A|--all}]
[{-n|--namespace} namespace]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

UI

To list installed applications:

1. In the navigation menu, select **Apps**.
The **Apps** section expands to display further options in the navigation menu.
2. Select **Installed**.

The **Installed** page is displayed with a table listing all the installed applications.

Showing Application Details

- [CLI](#)
- [UI](#)

CLI

Use the `ocne application show` command to show details about an application in a catalog. The syntax is:

```
ocne application show
[{-c|--computed}]
[{-d|--difference}]
[{-n|--namespace} namespace]
{-r|--release} name
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To show details about an application:

```
ocne application show --namespace kube-flannel --release flannel --computed
```

UI

To view details about an installed application:

1. In the navigation menu, select **Apps**.
The **Apps** section expands to display further options in the navigation menu.
 2. Select **Installed**.
The **Installed** page is displayed with a table showing release information for each installed application.
 3. For more details of a particular application release, navigate to the row for that application and select the link in the **Name** column.
A more detailed page of release history is displayed for that application.
 4. Select the **Values** icon.
A dialog box showing configuration values in YAML format is displayed for the application.
-

Installing an Application

- CLI
- UI

CLI

Use the `ocne application install` command to install an application from a catalog. The syntax is:

```
ocne application install
{-b|--built-in-catalog}
[{-c|--catalog} name]
{-N|--name} name
[{-n|--namespace} namespace]
{-r|--release} name
[{-u|--values} URI]
[{-v|--version} version]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To install an application from the default catalog:

```
ocne application install --release ingress-nginx --namespace ingress-nginx --
name ingress-nginx
```

UI

To install an application from the default catalog:

1. In the navigation menu, select **Apps**.

The **Applications** page is displayed listing all the applications available in the default catalog.

2. Enter characters in the **Search** field to display only applications whose name contain the entered characters.
3. Find the application you need to install and select **Install**.

A dialog box appears with the following fields:

- An edit box with YAML configuration for the application that's to be installed.
You can edit the YAML configuration according to the system requirements.
- The following mandatory fields: **Release Name**, **Namespaces**, **Versions** and **Release Description**.

4. Select **Install**.

The dialog is dismissed, and a message is displayed onscreen to confirm the installation request has been accepted.

The installation completes and a message is displayed confirming the release has been created successfully.

Updating an Application

- [CLI](#)
- [UI](#)

CLI

Use the `ocne application update` command to update an installed application. The syntax is:

```
ocne application update
{-b|--built-in-catalog}
[{-c|--catalog} name]
[{-n|--namespace} namespace]
{-r|--release} name
[--reset-values]
[{-u|--values} URI]
[{-v|--version} version]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To update an application with extra configuration information, use a values configuration file. For example, to update the `prometheus` application to include the Prometheus `node-exporter`:

```
ocne application update --release prometheus --namespace prometheus --values
- << EOF
serviceAccounts:
  nodeExporter:
    create: true
    name:
    annotations: {}
nodeExporter:
  enabled: true
  image:
    repository: container-registry.oracle.com/verrazzano/node-exporter
    tag: v1.3.1
    pullPolicy: IfNotPresent
EOF
```

UI

1. In the navigation menu, select **Apps**.
The **Apps** section expands to display further options in the navigation menu.
2. Select **Installed**.

The **Installed** page is displayed with a table listing the installed applications.

3. Go to the row for the application to be updated and select the link in the **Name** column.
A more detailed page of release history is displayed for that application.

4. Select **Upgrade**.

A dialog box appears with the following fields:

- **Release Description**

Enter a description for the release.

- **Versions**

Select a version from the list.

5. Select **Upgrade**.

The dialog is dismissed, and on-screen messages are displayed to confirm the upgrade request has been successfully submitted and completed.

The page with the application release history now includes the completed upgrade listed with the release description you entered in an earlier step.

Creating an Application Template

The configuration options for an application can be extracted from a catalog and viewed, saved to a file, or edited directly. To view the configuration, you generate a template. To save the template to a file to use for an installation, redirect the output to a file.

To edit a template interactively, set the `EDITOR` environment variable to the location of a text editing program. For example:

```
export EDITOR=/usr/bin/vim
```

-
- [CLI](#)
 - [UI](#)

CLI

Use the `ocne application template` command to create an application template. The syntax is:

```
ocne application template
{-c|--catalog} name
[{-i|--interactive}]
{-N|--name} name
[{-v|--version} version]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To display a template for an application named `prometheus` in the default catalog:

```
ocne application template --name prometheus
```

To save a template for an application to a file, pipe the output to a file:

```
ocne application template --name prometheus > mytemplate.yaml
```

To save a template for an application and edit it interactively:

```
ocne application template --name prometheus --interactive
```

The template is saved to a local file using the naming convention `application_name-values.yaml`.

UI

The following steps show you how to edit an application template in the UI before proceeding with the installation:

1. In the navigation menu, select **Apps**.

The **Apps** section expands to display further options in the navigation menu.

2. Select **Oracle Cloud Native Environment Application Catalog**.

The **Applications** page is displayed, listing the applications available in the catalog.

3. Select the **Install** option for the application whose template you're creating.

A dialog box appears with an edit box containing the application template for application you selected. The dialog also contains mandatory fields to configure the installation.

- a. Edit the **application template** as required.
- b. Complete the mandatory fields: **Release Name**, **Namespaces**, **Versions**, **Release Description**

4. Select **Install**.

The dialog is dismissed, and on-screen messages are displayed to confirm the installation request has been successfully submitted and completed.

Installing an Application from a Template

- CLI
- UI

CLI

Use the `ocne application install` command with an application template file as the values configuration file. The syntax is:

```
ocne application install
{-b|--built-in-catalog}
[{-c|--catalog} name]
{-N|--name} name
[{-n|--namespace} namespace]
{-r|--release} name
[{-u|--values} URI]
[{-v|--version} version]
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To install an application from a catalog using configuration information from an application template:

```
ocne application install --release prometheus --namespace prometheus --name
prometheus --values mytemplate.yaml
```

UI

To install an application from the default catalog:

1. In the navigation menu, select **Apps**.

The **Applications** page is displayed listing all the applications available in the default catalog.

2. Enter characters in the **Search** field to display only applications whose name contain the entered characters.
3. Find the application you need to install and select **Install**.

A dialog box appears with the following fields:

- An edit box with YAML configuration for the application that's to be installed.
Paste the contents of the application template file into the edit box.
- The following mandatory fields: **Release Name**, **Namespaces**, **Versions** and **Release Description**.

4. Select **Install**.

The dialog is dismissed, and a message is displayed onscreen to confirm the installation request has been accepted.

The installation completes and a message is displayed confirming the release has been created successfully.

Uninstalling an Application

- [CLI](#)
- [UI](#)

CLI

Use the `ocne application uninstall` command to uninstall an installed application. The syntax is:

```
ocne application uninstall  
[{-n|--namespace} namespace]  
{-r|--release} name
```

For more information on the syntax options, see [Oracle Cloud Native Environment: CLI](#).

Examples:

To uninstall an application:

```
ocne application uninstall --release prometheus --namespace prometheus
```

UI

To uninstall an application:

1. In the navigation menu, select **Apps**.
The **Apps** section expands to display further options in the navigation menu.
 2. Select **Installed**.
The **Installed** page is displayed with a table showing release information for each installed application.
 3. Go to the row for the application to be uninstalled and select the link in the **Name** column.
A more detailed page of release history is displayed for that application.
 4. Select **Delete**.
A dialog box appears asking you if you're sure you want to proceed.
 5. Select **YES** to proceed.
The dialog is dismissed, and a message is displayed onscreen to confirm the Delete request has been accepted.
The Uninstall completes and a message is displayed confirming the Delete request has been successfully completed.
 6. Confirm the deleted application is no longer listed in the **Installed** page (see [Showing Application Details](#) for information on how to view the installed applications).
-