

# Oracle Cloud Native Environment Updates and Upgrades for Release 1.9



F93856-01  
May 2024



Oracle Cloud Native Environment Updates and Upgrades for Release 1.9,  
F93856-01

Copyright © 2022, 2024, Oracle and/or its affiliates.

# Contents

## Preface

---

Documentation License	iv
Conventions	iv
Documentation Accessibility	iv
Access to Oracle Support for Accessibility	iv
Diversity and Inclusion	v

## 1 Introduction to Updating and Upgrading

---

## 2 Updating the Host OS

---

## 3 Updating to an Errata Release

---

Updating the Operator Node	3-1
Updating the Kubernetes Nodes	3-1
Updating the Kubernetes Cluster	3-2

## 4 Upgrading to Release 1.9

---

Changing the Software Packages Source	4-1
Oracle Linux 9	4-1
Oracle Linux 8	4-1
Upgrading the Operator Node	4-2
Upgrading the Kubernetes Nodes	4-3
Upgrading the Kubernetes Cluster	4-3
Upgrading the Calico Module	4-4
Upgrading the Oracle Cloud Infrastructure Cloud Controller Manager Module	4-5
Upgrading the Istio Module	4-5
Upgrading the KubeVirt Module	4-6

# Preface

This document contains information about updating or upgrading Oracle Cloud Native Environment.

## Documentation License

The content in this document is licensed under the [Creative Commons Attribution–Share Alike 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.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

## Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab>.

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

# 1

## Introduction to Updating and Upgrading

This document shows you how to *update* Oracle Cloud Native Environment and Kubernetes to the latest errata release, or *upgrade* from Release 1.8 to Release 1.9. This chapter uses the term *upgrade* to mean both upgrade and update as the overall process is the same.

The first step to upgrading is to upgrade the Oracle Cloud Native Environment software packages. This involves stopping the Platform API Server or Platform Agent on the node, upgrading the Oracle Cloud Native Environment packages, and restarting the Platform API Server or Platform Agent.

The next step is to upgrade the Kubernetes software packages. This is performed by the Platform API Server when you issue the appropriate `olcnectl module update` command.

You can upgrade a highly available cluster without bringing down the cluster. Control plane nodes are upgraded serially, so as one control plane node is taken offline, another control plane node takes control of the cluster. In a cluster with a single control plane node, the control plane node is offline for a short time while the upgrade is performed.

Worker nodes are also upgraded serially. If applications are running on more than one worker node, they remain up, and available during an upgrade.

### ! Important:

Certain Kubernetes rules might prevent a node from being taken offline for upgrade. A PodDisruptionBudget is one of these objects. For a node to be taken offline, increase the number of running pods to exceed the MinAvailable value. For more information about PodDisruptionBudgets see the upstream [Kubernetes documentation](#).

Before an upgrade begins, a back up is taken of the control plane nodes to help in any recovery that might be needed if a failure occurs.

### ! Important:

In the event of a module update failure, you can recover control plane nodes using the back up. For information on restoring from a control plane node back up, see [Kubernetes Module](#).

The Kubernetes release (either an errata or a new release) is then upgraded on each node. Control plane nodes are upgraded first, then the worker nodes. During the node upgrade process, the following steps are performed:

1. The node is drained (using the `kubectl drain` command) from the cluster, which evicts the pods.

2. The `kubeadm` package is upgraded.
3. The node is upgraded using the `kubeadm upgrade` command.
4. The `kubectl` and `kubelet` packages are upgraded.
5. The `kubelet` service is restarted.
6. The node is returned to the cluster (using the `kubectl uncordon` command) and is made available to run pods.

To update or upgrade Kubernetes, you update the Kubernetes module in an environment using the `olcnextl module update` command. The `olcnextl module update` command options shown in the following chapters are the minimum commands required to upgrade Kubernetes. You might also want to use these extra options:

- The `--generate-scripts` option generates scripts you can run for each node in the event of any validation failures during the update of the module. A script is created for each node in the module, saved to the local directory, and named `hostname:8090.sh`.
- The `--force` option suppresses the prompt displayed to confirm you want to update the module.
- The `--container-registry` option lets you specify a new container registry that becomes the default whenever running updates or upgrades. For example:  

```
--container-registry container-registry-austin-mirror.oracle.com/  
olcne/
```

# 2

## Updating the Host OS

To update the host OS on the nodes in an Oracle Cloud Native Environment, you must disable the Oracle Cloud Native Environment software package repositories when you perform the update. This avoids updates to the Oracle Cloud Native Environment software packages during the OS update process.

For example, if you're using the Oracle Linux yum server for system updates on Oracle Linux 9, update the host OS using:

```
sudo dnf update --disablerepo ol9_olcne19
```

If you're using Oracle Linux 8, use:

```
sudo dnf update --disablerepo ol8_olcne19
```



# 3

## Updating to an Errata Release

This section describes how to update Oracle Cloud Native Environment to an errata release. You can use this procedure to update Kubernetes to the latest errata release. This updates the environment within the same *major.minor* release. For example, Release 1.9 is updated to the latest available software.

Perform each step in this chapter to update an environment to an errata release.

### ! Important:

Don't change the ULN channel or Oracle Linux yum server repository. They must remain as you set them during the installation.

## Updating the Operator Node

Update the operator node with the new Oracle Cloud Native Environment software packages.

To update the operator node:

1. On the operator node, stop the `olcne-api-server` service:

```
sudo systemctl stop olcne-api-server.service
```

2. Update the Platform CLI, Platform API Server, and utilities packages.

```
sudo dnf upgrade olcnectl olcne-api-server olcne-utils
```

3. Start the `olcne-api-server` service:

```
sudo systemctl start olcne-api-server.service
```

## Updating the Kubernetes Nodes

Update the Kubernetes nodes with the new Oracle Cloud Native Environment packages.

On the operator node, use the `olcnectl environment update` command to update the Platform Agent on each Kubernetes node in the environment. This example updates the Platform Agent on all nodes in the `myenvironment` environment.

```
olcnectl environment update olcne \  
--environment-name myenvironment
```

The Platform Agent is updated and the service is restarted on each node.

# Updating the Kubernetes Cluster

Update the cluster to the latest Kubernetes errata release.

On the operator node, use the `olcnectl module update` command to update to the latest Kubernetes errata release. This example updates a Kubernetes module named `mycluster` in the `myenvironment` environment to the latest Kubernetes errata release.

```
olcnectl module update \  
--environment-name myenvironment \  
--name mycluster
```

If you're using the NGINX load balancer deployed by the Platform CLI, also update NGINX on the control plane nodes. You specify the location from which to pull the NGINX container image used to update NGINX using the `--nginx-image` option. For example, include this extra line in the `olcnectl module update` command to update NGINX from the Oracle Container Registry:

```
--nginx-image container-registry.oracle.com/olcne/nginx:1.17.7
```

## ! Important:

Ensure you update to the latest NGINX release for the Oracle Cloud Native Environment release. To get the version number of the latest NGINX container image, see [Release Notes](#).

You can optionally use the `--log-level` option to set the level of logging displayed in the command output. By default, error messages are displayed. For example, you can set the logging level to show all messages when you include:

```
--log-level debug
```

The log messages are also saved as an operation log. You can view operation logs as commands are running, or when they've completed. For more information using operation logs, see [Platform Command-Line Interface](#).

The nodes in the cluster are updated to the latest errata release and the cluster's health is validated.

# 4

## Upgrading to Release 1.9

This section describes how to upgrade Oracle Cloud Native Environment from Release 1.8 to Release 1.9.

When the Oracle Cloud Native Environment packages and the Kubernetes cluster are upgraded to Release 1.9, you can use the new features in this release.

Perform each step in this chapter to upgrade an environment from Release 1.8 to Release 1.9.

### ! Important:

Before you upgrade to Release 1.9, update to the latest errata release of Oracle Cloud Native Environment Release 1.8. For information on updating the environment, see [Updates and Upgrades for Release 1.8](#).

## Changing the Software Packages Source

This section contains information on setting the location for the software packages for the OS on which you want to upgrade the Oracle Cloud Native Environment software.

### Oracle Linux 9

If the OS on the nodes is Oracle Linux 9, update the package source using the information in this section.

If you're using the Oracle Linux yum server for system updates, on each node update the `oracle-olcne-release-el9` release package. Enable the `o19_olcne19` repository and disable the `o19_olcne18` and `o19_olcne17` repositories, which are for the previous releases.

On each node, run:

```
sudo dnf update oracle-olcne-release-el9
sudo dnf config-manager --enable o19_olcne19
sudo dnf config-manager --disable o19_olcne18 o19_olcne17
```

If the systems are registered to use ULN, use the ULN web interface to subscribe each system to the `o19_x86_64_olcne19` channel. Ensure you unsubscribe each system from the `o19_x86_64_olcne18` and `o19_x86_64_olcne17` channels, which are for the previous releases.

### Oracle Linux 8

If the OS on the nodes is Oracle Linux 8, update the package source using the information in this section.

If you're using the Oracle Linux yum server for system updates, on each node update the `oracle-olcne-release-el8` release package. Enable the `ol8_olcne19` repository and disable the repositories for the previous releases. Disable the following repositories:

- `ol8_olcne18`
- `ol8_olcne17`
- `ol8_olcne16`
- `ol8_olcne15`
- `ol8_olcne14`
- `ol8_olcne13`
- `ol8_olcne12`

On each node, run:

```
sudo dnf update oracle-olcne-release-el8
sudo dnf config-manager --enable ol8_olcne19
sudo dnf config-manager --disable ol8_olcne18 ol8_olcne17 ol8_olcne16
ol8_olcne15 ol8_olcne14 ol8_olcne13 ol8_olcne12
```

If the systems are registered to use ULN, use the ULN web interface to subscribe each system to the `ol8_x86_64_olcne19` channel. Ensure you unsubscribe each system from the following channels:

- `ol8_x86_64_olcne18`
- `ol8_x86_64_olcne17`
- `ol8_x86_64_olcne16`
- `ol8_x86_64_olcne15`
- `ol8_x86_64_olcne14`
- `ol8_x86_64_olcne13`
- `ol8_x86_64_olcne12`

## Upgrading the Operator Node

Upgrade the operator node with the new Oracle Cloud Native Environment software packages.

To upgrade the operator node on Oracle Linux:

1. On the operator node, stop the `olcne-api-server` service:

```
sudo systemctl stop olcne-api-server.service
```

2. Update the Platform CLI, Platform API Server, and utilities packages:

```
sudo dnf update olcnectl olcne-api-server olcne-utils
```

3. Start the `olcne-api-server` service:

```
sudo systemctl start olcne-api-server.service
```

## Upgrading the Kubernetes Nodes

Upgrade the Kubernetes nodes with the new Oracle Cloud Native Environment software packages.

On the operator node, use the `olcnectl environment update` command to upgrade the Platform Agent on each node in the environment. This example upgrades the Platform Agent on all nodes in the `myenvironment` environment.

```
olcnectl environment update olcne \  
--environment-name myenvironment
```

The Platform Agent is upgraded and the service is restarted on each node.

## Upgrading the Kubernetes Cluster

Upgrade the cluster to Kubernetes Release 1.29.3.

On the operator node, use the `olcnectl module update` command to upgrade to the latest Kubernetes release available for Oracle Cloud Native Environment Release 1.9. This example upgrades a Kubernetes module named `mycluster` in the `myenvironment` environment to Kubernetes Release 1.29.3.

```
olcnectl module update \  
--environment-name myenvironment \  
--name mycluster \  
--kube-version 1.29.3
```

The `--kube-version` option specifies the release to which you want to upgrade. This example uses release number 1.29.3.

### ! Important:

Ensure you upgrade to the latest Kubernetes release. To get the version number of the latest Kubernetes release for Oracle Cloud Native Environment Release 1.9, see [Release Notes](#).

You can optionally use the `--log-level` option to set the level of logging displayed in the command output. By default, error messages are displayed. For example, you can set the logging level to show all messages when you include:

```
--log-level debug
```

The log messages are also saved as an operation log. You can view operation logs as commands are running, or when they've completed. For more information using operation logs, see [Platform Command-Line Interface](#).

When each node in the cluster is upgraded to the latest Kubernetes release, the cluster's health is validated and the upgrade completes.

## Upgrading the Calico Module

If you have the Calico module installed, you must upgrade it.

### ! Important:

Ensure you upgrade to the latest releases for each component in this section. To get the version numbers of the latest releases for Oracle Cloud Native Environment Release 1.9, see [Release Notes](#).

To upgrade Calico, on the operator node, use the `olcnectl module update` command to upgrade to the latest Calico release available for Oracle Cloud Native Environment Release 1.9. This example upgrades an Calico module named `mycalico` in the `myenvironment` environment to Calico Release 3.27.0.

```
olcnectl module update \  
--environment-name myenvironment \  
--name mycalico \  
--calico-version 3.27.0
```

The `--calico-version` option specifies the release to which you want to upgrade. This example uses release number 3.27.0.

### ! Important:

Ensure you upgrade to the latest Calico release. To get the version number of the latest Calico release for Oracle Cloud Native Environment Release 1.9, see [Release Notes](#).

You can optionally use the `--log-level` option to set the level of logging displayed in the command output. By default, error messages are displayed. For example, you can set the logging level to show all messages when you include:

```
--log-level debug
```

The log messages are also saved as an operation log. You can view operation logs as commands are running, or when they've completed. For more information using operation logs, see [Platform Command-Line Interface](#).

# Upgrading the Oracle Cloud Infrastructure Cloud Controller Manager Module

If you have the Oracle Cloud Infrastructure Cloud Controller Manager module installed, you must also upgrade it.

To upgrade the Oracle Cloud Infrastructure Cloud Controller Manager module, on the operator node, use the `olcnectl module update` command. This example upgrades the module named `myoci` in the `myenvironment` environment to the Oracle Cloud Infrastructure Cloud Controller Manager module Release 1.28.0

```
olcnectl module update \  
--environment-name myenvironment \  
--name myoci \  
--oci-ccm-version 1.28.0 \  
--oci-container-registry "" \  
--ccm-container-registry ""
```

The `--oci-ccm-version` option specifies the release to which you want to upgrade. This example uses release number 1.28.0.

## ! Important:

Ensure you upgrade to the latest release of the Oracle Cloud Infrastructure Cloud Controller Manager module. To get the version numbers of the latest releases for Oracle Cloud Native Environment Release 1.9, see [Release Notes](#).

For registry options `--oci-container-registry` and `--ccm-container-registry`, the example assumes public registries are being used, and both options are therefore set to an empty string (`""`) so the Platform API Server automatically configures the correct public container registry from which to pull the Oracle Cloud Infrastructure Cloud Controller Manager container images. If, however, you're using private registries, and have already set values for `--oci-container-registry` and `--ccm-container-registry` during the installation process, then you don't have to specify these options.

You can optionally use the `--log-level` option to set the level of logging displayed in the command output. By default, error messages are displayed. For example, you can set the logging level to show all messages when you include:

```
--log-level debug
```

The log messages are also saved as an operation log. You can view operation logs as commands are running, or when they've completed. For more information using operation logs, see [Platform Command-Line Interface](#).

## Upgrading the Istio Module

If you have the Istio module installed, you must also upgrade it.

**! Important:**

Ensure you upgrade to the latest releases for each component in this section. To get the version numbers of the latest releases for Oracle Cloud Native Environment Release 1.9, see [Release Notes](#).

To upgrade Istio, on the operator node, use the `olcnectl module update` command to upgrade to the latest Istio release available for Oracle Cloud Native Environment Release 1.9. This example upgrades an Istio module named `myistio` in the `myenvironment` environment to Istio Release 1.20.4.

```
olcnectl module update \  
--environment-name myenvironment \  
--name myistio \  
--istio-version 1.20.4
```

The `--istio-version` option specifies the release to which you want to upgrade. This example uses release number 1.20.4.

When you upgrade to Istio Release 1.20.4, the update iterates through each Istio release up to Release 1.20.4. When each node in the cluster is upgraded to the next Istio release, the cluster's health is validated. If the cluster is healthy, the cycle of back up, upgrade to the next release, and cluster validation starts again, until all nodes are upgraded to the latest Istio release.

You can optionally use the `--log-level` option to set the level of logging displayed in the command output. By default, error messages are displayed. For example, you can set the logging level to show all messages when you include:

```
--log-level debug
```

The log messages are also saved as an operation log. You can view operation logs as commands are running, or when they've completed. For more information using operation logs, see [Platform Command-Line Interface](#).

## Upgrading the KubeVirt Module

If you have the KubeVirt module installed, you must also upgrade it.

To upgrade the KubeVirt module, on the operator node, use the `olcnectl module update` command. This example upgrades the module named `mykubevirt` in the `myenvironment` environment to the KubeVirt module Release 1.1.1

```
olcnectl module update \  
--environment-name myenvironment \  
--name mykubevirt \  
--kubevirt-version 1.1.1
```

The `--kubevirt-version` option specifies the release to which you want to upgrade. This example uses release number 1.1.1.



**!** **Important:**

Ensure you upgrade to the latest release of the KubeVirt module. To get the version numbers of the latest releases for Oracle Cloud Native Environment Release 1.9, see [Release Notes](#).

You can optionally use the `--log-level` option to set the level of logging displayed in the command output. By default, error messages are displayed. For example, you can set the logging level to show all messages when you include:

```
--log-level debug
```

The log messages are also saved as an operation log. You can view operation logs as commands are running, or when they've completed. For more information using operation logs, see [Platform Command-Line Interface](#).