

Oracle Cloud Native Environment Updates and Upgrades for Release 1.5



F53963-09
July 2023



Oracle Cloud Native Environment Updates and Upgrades for Release 1.5,
F53963-09

Copyright © 2022, 2023, Oracle and/or its affiliates.

Contents

Preface

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

1 Introduction to Updating and Upgrading

2 Updating the Host Operating System

3 Updating to an Errata Release

Updating the Operator Node	3-1
Updating the Kubernetes Nodes	3-2
Upgrading qemu-kvm	3-2
Updating the Kubernetes Cluster	3-3
Updating Istio	3-3

4 Upgrading to Release 1.5

Changing the Software Packages Source	4-1
Oracle Linux 8	4-1
Oracle Linux 7	4-2
Upgrading the Operator Node	4-2
Upgrading the Kubernetes Nodes	4-3
Upgrading the Kubernetes Cluster	4-3
Upgrading Istio	4-4
Upgrading the Oracle Cloud Infrastructure Container Storage Interface Module	4-5

5 Upgrading to Oracle Linux 8 using Leapp

Upgrading the Operator Node	5-2
Upgrading Kubernetes Nodes	5-4

Preface

! Important:

The software described in this documentation is either in Extended Support or Sustaining Support. See [Oracle Open Source Support Policies](#) for more information.

We recommend that you upgrade the software described by this documentation as soon as possible.

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

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	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.
monospace	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/>.

For information about the accessibility of the Oracle Help Center, see the Oracle Accessibility Conformance Report at <https://www.oracle.com/corporate/accessibility/templates/t2-11535.html>.

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

Important:

The software described in this documentation is either in Extended Support or Sustaining Support. See [Oracle Open Source Support Policies](#) for more information.

We recommend that you upgrade the software described by this documentation as soon as possible.

This document shows you how to *update* Oracle Cloud Native Environment and Kubernetes to the latest errata release, or *upgrade* from Release 1.4 to Release 1.5. 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 your applications are running on more than one worker node, they should remain up and available during an upgrade.

Note:

Certain Kubernetes rules may prevent a node from being taken offline for upgrade. A PodDisruptionBudget is one of these objects. To allow 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 documentation at:

<https://kubernetes.io/docs/concepts/workloads/pods/disruptions/#how-disruption-budgets-work>

Before an upgrade begins, a back up is taken of the control plane nodes to assist 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 [Container Orchestration](#).

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 `olcnectl module update` command. The `olcnectl module update` command options shown in the following chapters are the minimum commands required to upgrade Kubernetes. You may also want to use these additional options:

- The `--generate-scripts` option generates scripts you can run for each node in the event of any validation failures encountered 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 allows you to 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 Operating System

! Important:

The software described in this documentation is either in Extended Support or Sustaining Support. See [Oracle Open Source Support Policies](#) for more information.

We recommend that you upgrade the software described by this documentation as soon as possible.

If you want to update the host operating system on the nodes in your 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 operating system update process.

For example, if you are using the Oracle Linux yum server for system updates, update the host operating system using:

Oracle Linux 8:

```
sudo dnf update --disablerepo ol8_olcne15
```

Oracle Linux 7:

```
sudo yum update --disablerepo ol7_olcne15
```

3

Updating to an Errata Release

! Important:

The software described in this documentation is either in Extended Support or Sustaining Support. See [Oracle Open Source Support Policies](#) for more information.

We recommend that you upgrade the software described by this documentation as soon as possible.

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 1.5 is updated to the latest available software.

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

! Important:

Do not change the ULN channel or Oracle Linux yum server repository. They should 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.

On Oracle Linux 8 enter:

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

On Oracle Linux 7 enter:

```
sudo yum update 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.

Upgrading qemu-kvm

The Oracle Cloud Native Environment Release 1.5.3 errata update for Oracle Linux 8 requires upgrading the `qemu-kvm` packages.

If your hosts are running Oracle Linux 8, and you are updating to Oracle Cloud Native Environment Release 1.5.3, you must perform the steps in this section to upgrade `qemu-kvm`.

If you are updating from a release prior to 1.5.3 to a release after 1.5.3 (that is, not directly updating to Release 1.5.3), and you have not performed this step, you should also upgrade `qemu-kvm` using the steps below.

! Important:

The `qemu-kvm` packages must be upgraded if you are updating to a errata update after Release 1.5.3 and you have not upgraded them in the Release 1.5.3 errata update. That is, you skipped the errata update for Release 1.5.3.

Run the following commands on all Kubernetes worker and control plane nodes to upgrade the `qemu-kvm` packages.

To update `qemu-kvm`:

1. Enable the `ol8_kvm_appstream` application stream:

```
sudo dnf config-manager --set-enabled ol8_kvm_appstream
```

2. Enable the `virt:kvm_utils` module:

```
sudo dnf module enable -y virt:kvm_utils
```

3. Upgrade the `qemu-kvm` module:

```
sudo dnf update -y qemu-kvm
```

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 are using the NGINX load balancer deployed by the Platform CLI, you should 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 additional 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:

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

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

Updating Istio

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

Note:

Make sure you update to the latest release of Istio. To get the version number of the latest Istio release for Oracle Cloud Native Environment Release 1.5 see [Release Notes](#).

To update Istio:

On the operator node, use the `olcnectl module update` command to update to the latest Istio release available for Oracle Cloud Native Environment Release 1.5. This example updates an Istio module named `myistio` in the `myenvironment` environment to Istio Release 1.15.7.

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

```
--name myistio \  
--istio-version 1.15.7
```

The `--istio-version` option specifies the release to which you want to update. This example uses release number 1.15.7.

When you update Istio, the update iterates through each Istio release up to Release 1.15.7. When each node in the cluster is updated to the next Istio release, the cluster's health is validated. If the cluster is healthy, the cycle of back up, update to the next release, and cluster validation starts again, until all nodes are updated to the latest Istio release.

4

Upgrading to Release 1.5

! Important:

The software described in this documentation is either in Extended Support or Sustaining Support. See [Oracle Open Source Support Policies](#) for more information.

We recommend that you upgrade the software described by this documentation as soon as possible.

This section describes how to upgrade Oracle Cloud Native Environment from Release 1.4 to Release 1.5.

When the Oracle Cloud Native Environment packages and the Kubernetes cluster are upgraded to Release 1.5, you can use the new features in this release.

Perform each step in this chapter in order to upgrade your environment from Release 1.4 to Release 1.5.

Changing the Software Packages Source

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

Oracle Linux 8

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

If the systems are registered to use ULN, use the ULN web interface to subscribe each system to the `ol8_x86_64_olcne15` channel. Make sure you unsubscribe each system from the following channels:

- `ol8_x86_64_olcne14`
- `ol8_x86_64_olcne13`
- `ol8_x86_64_olcne12`

If you are using the Oracle Linux yum server for system updates, on each node update the `oracle-olcne-release-el8` release package. Enable the `ol8_olcne15` repository and disable the repositories for the previous releases. They are:

- `ol8_olcne14`
- `ol8_olcne13`
- `ol8_olcne12`

On each node, run:

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

Oracle Linux 7

If the operating system on the nodes is Oracle Linux 7, update the package source using the information in this section.

Disable any ULN channels or Oracle Linux yum server repositories for previous releases and enable the one for Release 1.5.

If the systems are registered to use ULN, use the ULN web interface to subscribe the systems to the `ol7_x86_64_olcne15` channel. Make sure you unsubscribe each system from the following channels.

- `ol7_x86_64_olcne14`
- `ol7_x86_64_olcne13`
- `ol7_x86_64_olcne12`
- `ol7_x86_64_olcne11`
- `ol7_x86_64_olcne`
- `ol7_x86_64_developer`

If you are using the Oracle Linux yum server for system updates, on each node update the `oracle-olcne-release-el7` release package. Enable the `ol7_olcne15` repository and disable the following repositories:

- `ol7_olcne14`
- `ol7_olcne13`
- `ol7_olcne12`
- `ol7_olcne11`
- `ol7_olcne`
- `ol7_developer`

On each node, run:

```
sudo yum update oracle-olcne-release-el7
sudo yum-config-manager --enable ol7_olcne15
sudo yum-config-manager --disable ol7_olcne14 ol7_olcne13 ol7_olcne12
ol7_olcne11 ol7_olcne ol7_developer
```

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 8:

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
```

To upgrade the operator node on Oracle Linux 7:

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 yum 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 1.24.15.

On the operator node, use the `olcnectl module update` command to upgrade to the latest Kubernetes release available for Oracle Cloud Native Environment Release 1.5. This example upgrades a Kubernetes module named `mycluster` in the `myenvironment` environment to Kubernetes Release 1.24.15.

```
olcnectl module update \  
--environment-name myenvironment \  
--name mycluster \  
--kube-version 1.24.15
```

The `--kube-version` option specifies the release to which you want to upgrade. This example uses release number 1.24.15.

! Important:

Make sure you upgrade to the latest Kubernetes release. To get the version number of the latest Kubernetes release for Oracle Cloud Native Environment Release 1.5, see [Release Notes](#).

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

Upgrading Istio

If you have the Istio module installed, you must also upgrade it and upgrade the supporting Prometheus and Grafana modules.

! Important:

Make sure 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 1.5, see [Release Notes](#).

To upgrade Istio:

1. On the operator node, use the `olcnectl module update` command to upgrade to the latest Istio release available for Oracle Cloud Native Environment Release 1.5. This example upgrades an Istio module named `myistio` in the `myenvironment` environment to Istio Release 1.15.7.

```
olcnectl module update \  
--environment-name myenvironment \  
--name myistio \  
--istio-version 1.15.7
```

The `--istio-version` option specifies the release to which you want to upgrade. This example uses release number 1.15.7.

When you upgrade to Istio Release 1.15.7, the update iterates through each Istio release up to Release 1.15.7. 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.

2. Upgrade to the latest Prometheus release available for Oracle Cloud Native Environment Release 1.5. This example upgrades the Prometheus module named `prometheus` (the name assigned by default by the Istio module) in the `myenvironment` environment to Prometheus Release 2.31.1.

```
olcnectl module update \  
--environment-name myenvironment \  
--name prometheus \  
--prometheus-version 2.31.1
```

3. Upgrade to the latest Grafana release available for Oracle Cloud Native Environment Release 1.5. This example upgrades the Grafana module named `grafana` (the name assigned by default by the Istio module) in the `myenvironment` environment to Grafana Release 7.5.15.

```
olcnectl module update \  
--environment-name myenvironment \  
--name grafana \  
--grafana-version 7.5.15
```

Upgrading the Oracle Cloud Infrastructure Container Storage Interface Module

If you have the Oracle Cloud Infrastructure Container Storage Interface module installed, you must also upgrade it. The Oracle Cloud Infrastructure Container Storage Interface module (`oci-csi`) has been renamed to the *Oracle Cloud Infrastructure Cloud Controller Manager* module (`oci-ccm`) in Release 1.5. The functionality of this module has been improved to add the ability to create and manage Oracle Cloud Infrastructure load balancers.

During the upgrade of the Oracle Cloud Native Environment Platform API Server, the module is automatically migrated from `oci-csi` to `oci-ccm`, although you still need to update some module options to correctly configure the module.

To upgrade the `oci-ccm` module, on the operator node, use the `olcnectl module update` command. This example upgrades the module named `myoci` in the `myenvironment` environment to the `oci-ccm` module Release 1.24.0

```
olcnectl module update \  
--environment-name myenvironment \  
--name myoci \  
--oci-ccm-version 1.24.0 \  
--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 1.24.0.

Important:

Make sure 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 1.5, 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 (`""`) to allow the Platform API Server to automatically configure the correct public container registry from which to pull the Oracle Cloud Infrastructure Cloud Controller Manager container images. If, however, you are using private registries, and have already set values for `--oci-container-registry` and `--ccm-container-registry` during the installation process, then you do not have to specify these options.

Tip:

If you want to use the Oracle Cloud Infrastructure Cloud Controller Manager module to set up Kubernetes application load balancers, you need to add some additional networking information to the module. For information on the required network settings and how to update the module, see [Application Load Balancers](#).

5

Upgrading to Oracle Linux 8 using Leapp

! Important:

The software described in this documentation is either in Extended Support or Sustaining Support. See [Oracle Open Source Support Policies](#) for more information.

We recommend that you upgrade the software described by this documentation as soon as possible.

This chapter describes how to upgrade nodes in an Oracle Cloud Native Environment from Oracle Linux 7 to Oracle Linux 8 using the Leapp utility. The Leapp utility upgrades the operating system to Oracle Linux 8 and maintains the Oracle Cloud Native Environment installation.

For more information about using Leap, see [Oracle Linux 8: Performing System Upgrades With Leapp](#).

! Important:

If you want to upgrade the nodes to Oracle Linux 8, you must first install or upgrade to Oracle Cloud Native Environment Release 1.5 on Oracle Linux 7.

Upgrade Overview

To upgrade control plane and worker nodes, you remove a node from the cluster, upgrade the operating system, and then add the node back into the cluster. The basic steps to upgrade your cluster from Oracle Linux 7 to Oracle Linux 8 are:

1. Upgrade the operator node (where the Platform API server is located).
2. Remove control plane and worker nodes(s) from the cluster and upgrade them using Leapp.
3. Replace the nodes in the cluster.

It is recommended that during the upgrade of a highly available Kubernetes cluster, the cluster should always have:

- A minimum of three control plane nodes. A minimum of five control plane nodes in the cluster during the upgrade is recommended.
- A minimum of three workers nodes.
- The workload in your cluster should have enough resources to allow you to remove 1/3 of your worker nodes. Make sure the cluster has enough resources to continue running your applications during the upgrade.

Oracle recommends that you do not run the cluster with mixed operating systems for longer than required to perform the upgrade.

Setting up a Remote Console

The Leapp utility requires some set up on each node before you perform the upgrade. The basics of this are included in this section. It is recommended that you read the Leapp documentation before you perform these steps as you may need to perform other steps on your nodes before you upgrade them. For more information about using Leap, see [Oracle Linux 8: Performing System Upgrades With Leapp](#).

Set up a method to connect remotely to the system console. This allows you to monitor the upgrade as it occurs over system reboots. For example:

- Oracle Cloud Infrastructure, use the Cloud Shell connection console.
- Oracle X86 servers, use Oracle Integrated Lights Out Manager (ILOM).
- Oracle Private Cloud Appliance, use the Instance Console Connection.
- Oracle Linux Kernel-based Virtual Machines (KVM), use the appropriate virtualization management software for your deployment, such as Cockpit Web Console, virt-viewer or virt-manager.

If you connect to the system using SSH in a terminal, you are disconnected while the upgrade is being performed and cannot log in again until the upgrade is completed.

Upgrading the Operator Node

This section shows you how to upgrade the operator node from Oracle Linux 7 to Oracle Linux 8 using Leapp. You should upgrade the operator node before you upgrade the nodes in the Kubernetes cluster.

To upgrade the operator node:

1. On the system you want to upgrade, update the operating system to the latest packages.

```
sudo yum update --disablerepo ol7_olcne15
```

2. Reboot the system.

```
sudo reboot
```

3. When the system restarts, install the Leapp upgrade software package:

```
sudo yum install -y leapp-upgrade --enablerepo ol7_leapp,ol7_latest
```

4. Grant the `root` user SSH login permissions. Edit the `/etc/ssh/sshd_config` file to remove the comment for the `CopyPermitRootLogin yes` entry.

```
sudo sed -i 's/#PermitRootLogin yes/PermitRootLogin yes/g' /etc/ssh/sshd_config
```

5. There may be preupgrade steps listed in the Leapp documentation that apply to your systems which are not listed here, such as unmounting any filesystems, or disabling Secure Boot. Understand and perform any further steps to prepare for an upgrade of the system as described in [Oracle Linux 8: Performing System Upgrades With Leapp](#).

6. On the node that you want to upgrade, run the Leapp upgrade preinstallation check script. This script checks whether the system is ready for the upgrade and generates a report you can review prior to upgrading.

If this is an Oracle Cloud Infrastructure instance, use:

```
sudo leapp preupgrade --oci --enablerepo ol8_olcne15
```

For all other Oracle Linux systems, use:

```
sudo leapp preupgrade --oraclelinux --enablerepo ol8_olcne15
```

The host is checked to make sure the upgrade can proceed, and an `answerfile` is generated for when you perform the upgrade.

7. Review the contents of the generated `answerfile` and resolve any issues.

```
sudo cat /var/log/leapp/answerfile
```

You can safely remove the section from the `answerfile` for the blocking module check for `remove_pam_pkcs11_module_check`.

```
sudo leapp answer --section remove_pam_pkcs11_module_check.confirm=True
```

More information is available on analyzing the Leapp `answerfile` in the Leapp documentation.

8. Review any issues that you may need to resolve with the upgrade by reviewing the Leapp upgrade report file.

```
sudo cat /var/log/leapp/leapp-report.txt
```

More information is available on analyzing the Leapp upgrade report in the Leapp documentation.

9. Run the `leapp preupgrade` command again to make sure all upgrade issues have been resolved.

If this is an Oracle Cloud Infrastructure instance, use:

```
sudo leapp preupgrade --oci --enablerepo ol8_olcne15
```

For all other Oracle Linux systems, use:

```
sudo leapp preupgrade --oraclelinux --enablerepo ol8_olcne15
```

The output should report that there are no blocking issues that need to be resolved.

10. You can review any risks with the upgrade by reading the upgrade report file.

```
sudo cat /var/log/leapp/leapp-report.txt
```

11. Run the Leapp upgrade script.

It is recommended that you run this upgrade commands using the remote console you set up or is available to your environment so you can monitor the upgrade as it occurs, and interact with the command line without losing the connection to the host. If you use an SSH terminal connection to the system, you can run the upgrade script, but your session is disconnected when the system is restarted.

If this is an Oracle Cloud Infrastructure instance, use:

```
sudo leapp upgrade --oci --enablerepo ol8_olcne15
```

For all other Oracle Linux systems, use:

```
sudo leapp upgrade --oraclelinux --enablerepo ol8_olcne15
```

A transaction check is performed and the output displayed.

12. Reboot the system to start the upgrade.

```
sudo reboot
```

13. The upgrade is performed and this may take some time. You can monitor the upgrade using the system console. When the system is upgraded you can log into the system again using SSH if you need.

When the upgrade is completed, reboot the system again.

```
sudo reboot
```

14. After the system restarts, you can confirm the operating system is upgraded using:

```
cat /etc/oracle-release
```

You should see the latest version of Oracle Linux 8 listed, for example:

```
Oracle Linux Server release 8.version
```

15. You can confirm the Oracle Cloud Native Environment software packages are updated using:

```
rpm -qa | egrep olcne | sort
```

The output should list the installed Oracle Cloud Native Environment software packages, which should now be for Oracle Linux 8, and look similar to:

```
olcne-api-server-version.el8.x86_64  
olcnectl-version.el8.x86_64  
olcne-selinux-version.el8.x86_64  
olcne-utils-version.el8.x86_64  
oracle-olcne-release-el8-version.el8.x86_64
```

Upgrading Kubernetes Nodes

This section shows you how to upgrade the Kubernetes nodes from Oracle Linux 7 to Oracle Linux 8 using Leapp.

You should upgrade the control plane nodes first, then the worker nodes. You need to scale out the nodes from the cluster, perform the upgrade to Oracle Linux 8, then scale the nodes back into the cluster.

Make sure you maintain high availability requirements for the cluster if required. For a highly available cluster, it is recommended that you have a cluster with at least five control plane nodes, and at least three worker nodes. The number of control plane nodes in a cluster must be an odd number equal to or greater than three, for example, 3, 5, or 7. As it is recommended to always have at least three control plane nodes in the cluster, with a recommended minimum of five nodes, this is likely to be the most common upgrade scenario.

It is recommended that you scale down the cluster to remove two control plane nodes at a time, then replace them with two upgraded Oracle Linux 8 nodes.

! Important:

Scaling down and up two nodes at a time, even if only one node needs to be upgraded to Oracle Linux 8, maintains a cluster quorum if a network outage occurs.

Worker nodes should be replaced in the cluster one at a time, to allow the applications running on the nodes to migrate to other nodes. If you have less than four worker nodes in the cluster, consider finding additional resources so that you do not enter a state where there are less than three worker nodes in the cluster. This may mean that you temporarily recruit another server to increase the worker node count to at least four during the upgrade process. Otherwise, your applications running on the worker nodes may not work as expected.

To upgrade the Kubernetes nodes:

1. On the operator node, use the `olcnectl module update` command to scale down the cluster to remove the Oracle Linux 7 node(s) you want to upgrade to Oracle Linux 8.

Use the following syntax to scale down (remove) control plane nodes:

```
olcnectl module update \
--environment-name environment_name \
--name cluster_name \
--master-nodes node_list
```

Use the following syntax to scale down worker nodes:

```
olcnectl module update \
--environment-name environment_name \
--name cluster_name \
--worker-nodes node_list
```

The `node_list` option should contain the list the nodes that should remain in the cluster, so omit the nodes to remove from this list.

💡 Tip:

You can see which nodes are in the cluster, by running the `kubectl get nodes` command on a control plane node where you have `kubectl set up`:

```
kubectl get nodes -o wide
```

2. On the system you want to upgrade, update the operating system to the latest packages.

```
sudo yum update --disablerepo ol7_olcnel5
```

3. Reboot the system.

```
sudo reboot
```

4. When the system restarts, install the Leapp upgrade software package:

```
sudo yum install -y leapp-upgrade --enablerepo ol7_leapp,ol7_latest
```

5. Grant the `root` user SSH login permissions. Edit the `/etc/ssh/sshd_config` file to remove the comment for the `CopyPermitRootLogin yes` entry.

```
sudo sed -i 's/#PermitRootLogin yes/PermitRootLogin yes/g' /etc/ssh/sshd_config
```

6. There may be preupgrade steps listed in the Leapp documentation that apply to your systems which are not listed here, such as unmounting any filesystems, or disabling Secure Boot. Understand and perform any further steps to prepare for an upgrade of the system as described in [Oracle Linux 8: Performing System Upgrades With Leapp](#).

7. On the node that you want to upgrade, run the Leapp upgrade preinstallation check script. This script checks whether the system is ready for the upgrade and generates a report you can review prior to upgrading.

If this is an Oracle Cloud Infrastructure instance, use:

```
sudo leapp preupgrade --oci --enablerepo ol8_olcne15,ol8_kvm_appstream
```

For all other Oracle Linux systems, use:

```
sudo leapp preupgrade --oraclelinux --enablerepo  
ol8_olcne15,ol8_kvm_appstream
```

The host is checked to make sure the upgrade can proceed, and an `answerfile` is generated for when you perform the upgrade.

8. Review the contents of the generated `answerfile` and resolve any issues.

```
sudo cat /var/log/leapp/answerfile
```

You can safely remove the section from the `answerfile` for the blocking module check for `remove_pam_pkcs11_module_check`.

```
sudo leapp answer --section remove_pam_pkcs11_module_check.confirm=True
```

More information is available on analyzing the Leapp `answerfile` in the Leapp documentation.

9. Review any issues that you may need to resolve with the upgrade by reviewing the Leapp upgrade report file.

```
sudo cat /var/log/leapp/leapp-report.txt
```

More information is available on analyzing the Leapp upgrade report in the Leapp documentation.

10. Run the `leapp preupgrade` command again to make sure all upgrade issues have been resolved.

If this is an Oracle Cloud Infrastructure instance, use:

```
sudo leapp preupgrade --oci --enablerepo ol8_olcne15,ol8_kvm_appstream
```

For all other Oracle Linux systems, use:

```
sudo leapp preupgrade --oraclelinux --enablerepo  
ol8_olcne15,ol8_kvm_appstream
```

The output should report that there are no blocking issues that need to be resolved.

11. Run the Leapp upgrade script.

It is recommended that you run this upgrade commands using the remote console you set up or is available to your environment so you can monitor the upgrade as it occurs, and interact with the command line without losing the connection to the

host. If you use an SSH terminal connection to the system, you can run the upgrade script, but your session is disconnected when the system is restarted.

If this is an Oracle Cloud Infrastructure instance, use:

```
sudo leapp upgrade --oci --enablerepo ol8_olcne15,ol8_kvm_appstream
```

For all other Oracle Linux systems, use:

```
sudo leapp upgrade --oraclelinux --enablerepo ol8_olcne15,ol8_kvm_appstream
```

A transaction check is performed and the output displayed.

12. Reboot the system to start the upgrade.

```
sudo reboot
```

13. The upgrade is performed and this may take some time. You can monitor the upgrade using the system console. When the system is upgraded you can log into the system again using SSH if you need.

When the upgrade is completed, reboot the system again.

```
sudo reboot
```

14. After the system restarts, you can confirm the operating system is upgraded using:

```
cat /etc/oracle-release
```

You should see the latest version of Oracle Linux 8 listed, for example:

```
Oracle Linux Server release 8.version
```

15. You can confirm the Oracle Cloud Native Environment software packages are updated using:

```
rpm -qa | egrep 'olcne|kube|common|kata|cri-o|kvm' | sort
```

The output should list the installed Oracle Cloud Native Environment software packages, which should now be for Oracle Linux 8, and look similar to:

```
common-version.el8.x86_64  
cri-o-version.el8.x86_64  
kata-version.el8.x86_64  
...
```

16. After the upgrade, SELinux is set to `permissive`. If SELinux was set to `enforcing` before the upgrade, update the SELinux configuration file to set it to `enforcing` again:

```
sudo sed -i s/^SELINUX=.*$/SELINUX=enforcing/ /etc/selinux/config; sudo setenforce 1
```

17. Run the post upgrade script:

```
sudo /etc/olcne/scripts/olcne-ol8-post-leapp-upgrade
```

18. On the operator node, use the `olcnectl module update` command to scale up the cluster to add the node(s) you have upgraded to Oracle Linux 8.

Use the following syntax to scale up (add) control plane nodes:

```
olcnectl module update \  
--environment-name environment_name \  
--name cluster_name \  
--master-nodes node_list
```

Use the following syntax to scale up worker nodes:

```
olcnectl module update \  
--environment-name environment_name \  
--name cluster_name \  
--worker-nodes node_list
```

The *node_list* option should contain the list the nodes that should be in the cluster.

 **Tip:**

You can see which nodes are in the cluster, by running the `kubectl get nodes` command on a control plane node where you have `kubectl` set up:

```
kubectl get nodes -o wide
```