# Oracle® Cloud Native Environment

## Updates and Upgrades for Release 1.2

F40898-05
November 2021

---

**Oracle Legal Notices**

# Table of Contents

# Preface

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

Document generated on: 2021-11-17 (revision: 1158)

## Audience

This document is written for system administrators who want to update or upgrade Oracle Cloud Native Environment. It is assumed that readers have a general understanding of the Oracle Linux operating system and container concepts.

## Related Documents

The latest version of this document and other documentation for this product are available at:

https://docs.oracle.com/en/operating-systems/olcne/

## 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. |
| *italic* | 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/.

## 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.

# Chapter 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 previous releases to Release 1.2. 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/
  ```

# Chapter 2 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 1.1 is updated to the latest available software, or Release 1.2 is updated to the latest 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.

## 2.1 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 7 enter:

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

   On Oracle Linux 8 enter:

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

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

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

## 2.2 Updating the Kubernetes Nodes

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

**To update the Kubernetes nodes:**

1. On the node to update, stop the `olcne-agent` service:

```
sudo systemctl stop olcne-agent.service
```

2. Update the Platform Agent and utilities packages.

   On Oracle Linux 7 enter:

```
sudo yum update olcne-agent olcne-utils
```

   On Oracle Linux 8 enter:

```
sudo dnf upgrade olcne-agent olcne-utils
```

3. Start the `olcne-agent` service:

```
sudo systemctl start olcne-agent.service
```

## 2.3 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.

**Important**

If you are updating from Release 1.2.0 to 1.2.2, or from Release 1.1.*x* to 1.1.10, you must also update your cluster to configure access to `externalIPs`. For information on the steps required, see Chapter 5, *Updating to Release 1.2.2 and 1.1.10*.

# Chapter 3 Upgrading to Release 1.1

This section describes how to upgrade Oracle Cloud Native Environment from Release 1.0 to Release 1.1.

When the Oracle Cloud Native Environment packages and the Kubernetes cluster are upgraded to Release 1.1, you can install the new Istio module and use other new features in this release such as scaling a Kubernetes cluster.

Perform each step in this chapter in order to upgrade your environment from Release 1.0 to Release 1.1.

## 3.1 Updating the Network Configuration

On each Kubernetes node, make changes to the firewall to set new rules to disable masquerading and to add the `cni0` interface to the trusted zone.

```
sudo firewall-cmd --remove-masquerade --permanent
sudo firewall-cmd --zone=trusted --add-interface=cni0 --permanent
```

## 3.2 Changing the Software Packages Source

Disable the ULN channel or Oracle Linux yum server repository for Release 1.0 and enable the one for Release 1.1.

If the systems are registered to use ULN, use the ULN web interface to subscribe the systems to the `ol7_x86_64_olcne11` channel. Make sure you unsubscribe each system from the following channels.

- `ol7_x86_64_olcne`
- `ol7_x86_64_olcne12`
- `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, disable the `ol7_olcne` and `ol7_developer` repositories, and enable the `ol7_olcne11` repository. On each node, run:

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

## 3.3 Upgrading the Operator Node

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

**To upgrade 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 yum update olcnectl olcne-api-server olcne-utils
   ```

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

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

## 3.4 Upgrading the Kubernetes Nodes

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

**To upgrade the Kubernetes nodes:**

1. On the node to update, stop the `olcne-agent` service:

```
sudo systemctl stop olcne-agent.service
```

2. Update the Platform Agent and utilities packages:

```
sudo yum update olcne-agent olcne-utils
```

3. Start the `olcne-agent` service:

```
sudo systemctl start olcne-agent.service
```

## 3.5 Upgrading the Kubernetes Cluster

Upgrade the cluster to Kubernetes Release 1.17.

On the operator node, use the `olcnectl module update` command to upgrade to the latest Kubernetes release available for Oracle Cloud Native Environment Release 1.1. This example upgrades a Kubernetes module named `mycluster` in the `myenvironment` environment to Kubernetes Release 1.17.

```
olcnectl module update \
--environment-name myenvironment \
--name mycluster \
--kube-version 1.17.9
```

The `--kube-version` option specifies the release to which you want to upgrade. This example uses release number 1.17.9.

**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 1.1, see *Release Notes*.

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

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

**Important**

Make sure you upgrade to the latest NGINX release. To get the version number of the latest NGINX container image for Oracle Cloud Native Environment Release 1.1, see *Release Notes*.

When you upgrade from Kubernetes Release 1.14 to 1.17, the update iterates through each Kubernetes release up to Release 1.17. That is, the nodes are upgraded to Kubernetes Release 1.15, then 1.16, and finally to 1.17.

**Important**

The Kubernetes Releases 1.15 and 1.16 should not be used, other than to perform the upgrade to Release 1.17.

When each node in the cluster is upgraded to the next Kubernetes 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 release.

# Chapter 4 Upgrading to Release 1.2

This section describes how to upgrade Oracle Cloud Native Environment from Release 1.1 to Release 1.2.

**Important**

If you want to upgrade from Release 1.0, you must first upgrade to Release 1.1, then upgrade to Release 1.2. For information on upgrading from Release 1.0 to Release 1.2, see Chapter 3, *Upgrading to Release 1.1*.

When the Oracle Cloud Native Environment packages and the Kubernetes cluster are upgraded to Release 1.2, you can use the new features in this release.

Perform each step in this chapter in order to upgrade your environment from Release 1.1 to Release 1.2.

## 4.1 Changing the Software Packages Source

Disable any ULN channels or Oracle Linux yum server repositories for previous releases and enable the one for Release 1.2.

If the systems are registered to use ULN, use the ULN web interface to subscribe the systems to the `ol7_x86_64_olcne12` channel. Make sure you unsubscribe each system from the following channels.

- `ol7_x86_64_olcne`
- `ol7_x86_64_olcne11`
- `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_olcne12` repository and disable the following repositories:

- `ol7_olcne`
- `ol7_olcne11`
- `ol7_developer`

On each node, run:

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

## 4.2 Upgrading the Operator Node

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

**To upgrade 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 yum update olcnectl olcne-api-server olcne-utils
```

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

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

## 4.3 Upgrading the Kubernetes Nodes

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

**To upgrade the Kubernetes nodes:**

1. On the node to update, stop the `olcne-agent` service:

```
sudo systemctl stop olcne-agent.service
```

2. Update the Platform Agent and utilities packages:

```
sudo yum update olcne-agent olcne-utils
```

3. Start the `olcne-agent` service:

```
sudo systemctl start olcne-agent.service
```

## 4.4 Upgrading the Kubernetes Cluster

Upgrade the cluster to Kubernetes Release 1.18.

On the operator node, use the `olcnectl module update` command to upgrade to the latest Kubernetes release available for Oracle Cloud Native Environment Release 1.2. This example upgrades a Kubernetes module named `mycluster` in the `myenvironment` environment to Kubernetes Release 1.18.

```
olcnectl module update \
--environment-name myenvironment \
--name mycluster \
--kube-version 1.18.18
```

The `--kube-version` option specifies the release to which you want to upgrade. This example uses release number 1.18.18.

**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 1.2, see *Release Notes*.

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

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

**Important**

Make sure you upgrade to the latest NGINX release. To get the version number of the latest NGINX container image for Oracle Cloud Native Environment Release 1.2, 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.

## 4.5 Upgrading Istio

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

On the operator node, use the `olcnectl module update` command to upgrade to the latest Istio release available for Oracle Cloud Native Environment Release 1.2. This example upgrades an Istio module named `myistio` in the `myenvironment` environment to Istio Release 1.9.8.

```
olcnectl module update \
--environment-name myenvironment \
--name myistio \
--istio-version 1.9.8
```

The `--istio-version` option specifies the release to which you want to upgrade. This example uses release number 1.9.8.

**Important**

Make sure you upgrade to the latest Istio release. To get the version number of the latest Istio release for Oracle Cloud Native Environment Release 1.2, see *Release Notes*.

When you upgrade from Istio Release 1.4.10 to 1.9.8, the update iterates through each Istio release up to Release 1.9.8. That is, the nodes are upgraded to Istio Release 1.5.10, then 1.6.12, and finally to 1.9.8.

**Important**

The Istio Releases 1.5.10 and 1.6.12 should not be used, other than to perform the upgrade to Release 1.9.8.

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 release.

# Chapter 5 Updating to Release 1.2.2 and 1.1.10

This section describes the additional configuration required when you update Oracle Cloud Native Environment from Release 1.2.0 to Release 1.2.2, and from Release 1.1.*x* to 1.1.10. Although these are errata updates, there is functionality that has changed in these releases related to the deployment of Kubernetes modules and a new Kubernetes service to validate `externalIPs`. For information on the changes in the Kubernetes module related to `externalIPs`, see *Container Orchestration*.

Kubernetes clusters that are updated to Releases 1.2.2 and 1.1.10 are able to use the new feature to restrict `externalIPs` in Kubernetes services. Updated clusters have a new configuration option set, `--restrict-service-externalip=false`.

**Important**

It is recommended to upgrade existing clusters to turn this feature on.

To enable this feature for an existing updated cluster, you need modify the Kubernetes module for the cluster with the required information to deploy the `externalip-validation-webhook-service` service.

**Important**

Any new Kubernetes modules you create in Release 1.2.2 and 1.1.10 onwards have the setting of `--restrict-service-externalip=true`, which means new clusters must be configured with the `externalIP` feature by default, unless you specify `--restrict-service-externalip=false` when creating the module.

**To deploy the service on an updated cluster:**

1. The `externalip-validation-webhook-service` Kubernetes service requires X.509 certificates be set up prior to updating the Kubernetes module. You can use certificates generated by Vault, your own certificates, or generate certificates using the `gen-certs-helper.sh` script. For information on setting up these certificates, see *Getting Started*.

2. On the operator node, use the `olcnectl module update` command to update the Kubernetes module to turn on this feature and provide the required certificate information. The `--restrict-service-externalip=true` option enables this feature. The other options set the location of the certificates. For example:

   ```
   olcnectl module update \
   --environment-name myenvironment \
   --name mycluster \
   --restrict-service-externalip=true \
   --restrict-service-externalip-ca-cert=/etc/olcne/configs/certificates/restrict_external_ip/production/c
   --restrict-service-externalip-tls-cert=/etc/olcne/configs/certificates/restrict_external_ip/production/
   --restrict-service-externalip-tls-key=/etc/olcne/configs/certificates/restrict_external_ip/production/n
   ```

   The `externalip-validation-webhook-service` Kubernetes service is deployed to the cluster.

# Chapter 6 Upgrading to Oracle Linux 8

This chapter describes how to upgrade nodes in an Oracle Cloud Native Environment from Oracle Linux 7 to Oracle Linux 8.

**Important**

If you want to upgrade the nodes to Oracle Linux 8, you must first install or upgrade to Oracle Cloud Native Environment Release 1.2 on Oracle Linux 7.

You cannot upgrade the operating system from Oracle Linux 7 to Oracle Linux 8 on the Oracle Cloud Native Environment nodes. You can either create a new set of nodes running Oracle Linux 8 and swap them for the existing Oracle Linux 7 nodes, or remove a node from the cluster, upgrade it, and then add it back to the cluster. The method you choose depends on how many server resources you have in your environment.

It is recommended that during the upgrade your Kubernetes 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.

The basic steps to upgrade your cluster from Oracle Linux 7 to Oracle Linux 8 are:

1. Upgrade the operator node.
2. Prepare new Oracle Linux 8 nodes if possible. If you do not have the server resources to use new nodes, you can upgrade the nodes as they are removed from the cluster.
3. If the cluster uses an external load balancer, add any new nodes to it.
4. Replace the nodes in the cluster.

## 6.1 Upgrading the Operator Node

This section shows you how to migrate the operator node from Oracle Linux 7 to Oracle Linux 8.

**To upgrade the operator node:**

1. Install Oracle Linux 8 on a new node and prepare it to use as an operator node. For information on preparing a node to use as an operator node, see *Getting Started*.
2. Copy the Oracle Cloud Native Environment data from the `/var/olcne` directory of the Oracle Linux 7 operator node to the same location on the new Oracle Linux 8 node. Alternatively, if you are using external storage for the `/var/olcne` directory, mount the shared storage on the new node.
3. Set up the X.509 certificate for the new node and start the Platform API Server service.

- If you are using Vault to authenticate nodes in the cluster, start the Platform API Server service using the Vault token. For information on starting the Platform API Server service with a Vault token, see *Getting Started*.
- If you are using CA certificates to authenticate nodes in the cluster, copy the certificate information to the node and start the Platform API Server service using the certificate. For information on starting the Platform API Server service with a CA certificate, see *Getting Started*.
- If you are using private CA certificates to authenticate the nodes in the cluster, copy the private key and the CA certificate used to generate new certificates for nodes from the Oracle Linux 7 node to the new Oracle Linux 8 node. The default location for these is:

  ```
  /path/configs/certificates/production/ca.cert
  /path/configs/certificates/production/ca.key
  ```

  Where *path* is the directory from which you originally ran the `gen-certs-helper.sh` script (it is most likely `/etc/olcne`).

  On the new Oracle Linux 8 node, generate a new certificate for the new operator node using the copied certificate and key, and copy the new certificate information to the node itself. For more information on generating and transferring certificates to nodes, see *Getting Started*.

  Start the Platform API Server service using the private certificate. For information on starting the Platform API Server with a CA certificate, see *Getting Started*.

4. Restart the Platform API Server.

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

## 6.2 Preparing an Oracle Linux 8 Node

This section shows you how to prepare an Oracle Linux 8 node to use as a Kubernetes node in the cluster. When the node is prepared it can be scaled into the cluster.

If you have the server resources, you can prepare a whole new set of nodes to add to the cluster in advance, or you can prepare the nodes as you remove them from an Oracle Linux 7 cluster.

**To prepare an Oracle Linux 8 node:**

1. Install Oracle Linux 8 on the node and prepare it to use as a Kubernetes node. For information on preparing a node to use as a Kubernetes node, see *Getting Started*.
2. Set up the X.509 certificate for the new node and start the Platform Agent service.
   - If you are using Vault to authenticate nodes in the cluster, start the Platform Agent service using the Vault token. For information on starting the Platform Agent service with a Vault token, see *Getting Started*.
   - If you are using CA certificates to authenticate nodes in the cluster, copy the certificate information to the node and start the Platform Agent service using the certificate. For information on starting the Platform Agent service with a CA certificate, see *Getting Started*.
   - If you are using private CA certificates to authenticate the nodes in the cluster, generate a new certificate and copy the new certificate information to the node. For more information on generating and transferring certificates to nodes, see *Getting Started*.

Start the Platform Agent service using the private certificate. For information on starting the Platform Agent with a CA certificate, see *Getting Started*.

## 6.3 Adding New Nodes to Your Load Balancer

If you are using an external load balancer for the Kubernetes cluster (set with the `--load-balancer` option when you created the Kubernetes module), add any new control plane nodes to it. If you are reusing nodes and keeping the same IP address and hostname, you do not need to add them to the load balancer, only any news nodes. If you are creating any new worker nodes and you have the Istio module deployed, you should also add them to your load balancer if required.

If you are using an Oracle Cloud Infrastructure load balancer, add any new control plane nodes to the appropriate backend set and set the port for the control plane nodes to `6443`. If you are using Istio with a load balancer, also add any new worker nodes to the the appropriate backend set.

If you are using the load balancer deployed by the Platform CLI (set with the `--virtual-ip` option when you created the Kubernetes module), you do not need to add the control plane nodes to it. This is done automatically when you scale the nodes into the cluster.

## 6.4 Upgrading the Kubernetes Cluster

This section shows you how to replace the nodes in a Kubernetes cluster to upgrade them from Oracle Linux 7 to Oracle Linux 8. The first option in this section upgrades the operating system in the cluster using the recommended cluster size of five control plane nodes and three worker nodes. If you do not have this recommended cluster size, alternative options are provided for different cluster types.

### 6.4.1 Replacing the Nodes

It is recommended that you have a cluster with at least five control plane nodes, and at least three worker nodes. This section shows you how to replace the Kubernetes cluster nodes in this scenario, with the addition of one extra worker node so that the minimum worker nodes requirement is maintained.

You can also use this method if you have more than five control plane nodes and more than 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 of the five control plane nodes, then replace them with two Oracle Linux 8 nodes. You can either use new nodes you prepared in advance, or upgrade the two removed nodes and scale them back in, depending on your server resources.

**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 replace the Kubernetes cluster nodes:**

1. From a control plane node, use the `kubectl get nodes` command to see the control plane and worker nodes in the cluster. In this example, there are five control plane and four worker nodes running Oracle Linux 7.

```
kubectl get nodes

NAME                   STATUS   ROLE     AGE     VERSION
control1.example.com   Ready    master   26h     v1.18.x+x.x.x.el7
control2.example.com   Ready    master   26h     v1.18.x+x.x.x.el7
control3.example.com   Ready    master   26h     v1.18.x+x.x.x.el7
control4.example.com   Ready    master   26h     v1.18.x+x.x.x.el7
control5.example.com   Ready    master   26h     v1.18.x+x.x.x.el7
worker1.example.com    Ready    <none>   26h     v1.18.x+x.x.x.el7
worker2.example.com    Ready    <none>   26h     v1.18.x+x.x.x.el7
worker3.example.com    Ready    <none>   26h     v1.18.x+x.x.x.el7
worker4.example.com    Ready    <none>   26h     v1.18.x+x.x.x.el7
```

2. On the operator node, scale down the cluster to remove two Oracle Linux 7 control plane nodes. This example removes the `control1.example.com` and `control2.example.com` nodes.

```
olcnectl module update \
--environment-name myenvironment \
--name mycluster \
--master-nodes control3.example.com:8090,control4.example.com:8090,control5.example.com:8090
```

   The cluster now has the minimum required number of control plane nodes, which is three.

   Allocate two of the new Oracle Linux 8 nodes, or upgrade the two nodes that have been removed from the cluster. For information on preparing nodes, see Section 6.2, “Preparing an Oracle Linux 8 Node”.

   Scale up the cluster to add the two Oracle Linux 8 control plane nodes. This example adds the `control1-ol8.example.com` and `control2-ol8.example.com` nodes as control plane nodes.

```
olcnectl module update \
--environment-name myenvironment \
--name mycluster \
--master-nodes control1-ol8.example.com:8090,control2-ol8.example.com:8090,control3.example.com:8090,contro
```

   The cluster now has the recommended number of control plane nodes, which is five.

   Repeat this process until all Oracle Linux 7 control plane nodes are replaced with Oracle Linux 8 nodes.

   As there are an odd number of control plane nodes in the cluster, there will be one final node that has not been upgraded. You should scale down and up two control plane nodes at a time, even though one of the nodes is already upgraded to Oracle Linux 8. This maintains an odd number of control plane nodes.

3. When all control plane nodes in the cluster are replaced with Oracle Linux 8 nodes, you can do the same for the worker nodes. Replace the worker nodes one at a time.

   Scale down the cluster to remove an Oracle Linux 7 worker node. This example removes the `worker1.example.com` node.

```
olcnectl module update \
--environment-name myenvironment \
--name mycluster \
--worker-nodes worker2.example.com:8090,worker3.example.com:8090,worker4.example.com:8090
```

There are three worker nodes left in the cluster, which is the recommended minimum.

Allocate one of the new Oracle Linux 8 nodes, or upgrade the node that has been removed from the cluster.

Scale up the cluster to add the Oracle Linux 8 worker node. This example adds the `worker1-ol8.example.com` node as a worker node.

```
olcnectl module update \
--environment-name myenvironment \
--name mycluster \
--worker-nodes worker1-ol8.example.com:8090,worker2.example.com:8090,worker3.example.com:8090,worker4.e
```

Repeat this process until all Oracle Linux 7 worker nodes are replaced with the new Oracle Linux 8 nodes.

4. On a control plane node, use the `kubectl get nodes` command to verify the cluster contains the new Oracle Linux 8 control plane and worker nodes and all the Oracle Linux 7 nodes have been removed. For example:

```
kubectl get nodes

NAME                       STATUS  ROLE     AGE     VERSION
control1-ol8.example.com   Ready   master   20m34s  v1.18.x+x.x.x.el8
control2-ol8.example.com   Ready   master   18m12s  v1.18.x+x.x.x.el8
control3-ol8.example.com   Ready   master   14m12s  v1.18.x+x.x.x.el8
control4-ol8.example.com   Ready   master   14m12s  v1.18.x+x.x.x.el8
control5-ol8.example.com   Ready   master   14m12s  v1.18.x+x.x.x.el8
worker1-ol8.example.com    Ready   <none>   9m40s   v1.18.x+x.x.x.el8
worker2-ol8.example.com    Ready   <none>   6m28s   v1.18.x+x.x.x.el8
worker3-ol8.example.com    Ready   <none>   4m28s   v1.18.x+x.x.x.el8
worker4-ol8.example.com    Ready   <none>   1m28s   v1.18.x+x.x.x.el8
```

## 6.4.2 Replacing the Nodes in a Three Control Plane Node Cluster

If you have a cluster with three control plane nodes, you should create an extra two Oracle Linux 8 control plane nodes using new servers to use during the upgrade. You add these two new control plane nodes to the cluster then remove two Oracle Linux 7 control plane nodes to upgrade the cluster. You repeat this until all control plane nodes are replaced. This means you can maintain the minimum of three control plane nodes in the cluster during the upgrade.

**Tip**

If you do not have servers to use as a temporary control plane nodes, you could use worker nodes. Scale down the cluster to remove two worker nodes, prepare them as Oracle Linux 8 nodes, and scale up the cluster to add them as a control plane nodes. When the upgrade is completed, remove the two control plane nodes and add them back into the cluster as worker nodes.

The basic steps in this process are below. For details of these steps, see Section 6.4.1, “Replacing the Nodes”.

**To upgrade a three control plane node cluster:**

1. Create two Oracle Linux 8 nodes. The nodes are used as a temporary control plane nodes during the upgrade to maintain a cluster quorum.
2. Scale up the cluster to add the two new nodes as control plane nodes. The cluster now has five control plane nodes.
3. Scale down the cluster to remove two Oracle Linux 7 control plane nodes. The cluster now has three control plane nodes.
4. Upgrade the removed nodes to Oracle Linux 8 and prepare them to use as a new nodes.
5. Repeat the scaling up and down until all control plane nodes are replaced with Oracle Linux 8 nodes.
6. Upgrade the worker nodes by scaling down the cluster to remove a worker node, upgrading it, then scaling up to add it into the cluster again.

## 6.4.3 Replacing the Nodes in a Single Control Plane Node Cluster

If you have cluster with one control plane node and a load balancer, use the same procedure as described in Section 6.4.2, “Replacing the Nodes in a Three Control Plane Node Cluster”. You do not need to create two extra control plane nodes in this situation, one extra control plane node is sufficient. A single control plane node cluster does not meet the requirements for a cluster quorum, but creating an extra control plane node to add to the cluster while you upgrade the original node means your cluster remains up during the upgrade.

If you have a cluster with one control plane node and you have not set a load balancer, perform the following steps. During the upgrade the cluster is taken offline while the control plane node is replaced.

**Important**

During the upgrade you must use the same IP address and hostname for the control plane node you used when creating the cluster.

**To migrate a single node cluster with no load balancer:**

1. If the Platform API Server is also on the control plane node, you must migrate it to another node. For information on migrating the Platform API Server, see Section 6.1, “Upgrading the Operator Node”.
2. On the operator node, back up the cluster using the `olcnectl module backup` command. For example:

```
olcnectl module backup --environment-name myenvironment --name mycluster
```

3. Upgrade the control plane node to Oracle Linux 8. For information on preparing nodes, see Section 6.2, “Preparing an Oracle Linux 8 Node”.
4. On the operator node, restart the Platform API Server.

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

5. Validate the nodes are set up correctly using the `olcnectl module validate` command. For example:

```
olcnectl module validate --environment-name myenvironment --name mycluster
```

6. Restore the cluster using the `olcnectl module restore` command. For example:

```
olcnectl module restore --environment-name myenvironment --name mycluster
```

7. When the control plane node is replaced with an Oracle Linux 8 node, you can replace the worker nodes. Details of replacing worker nodes is provided in Section 6.4.1, “Replacing the Nodes”.