

Oracle Utilities Live Energy Connect

Installation Guide

F74938-06

Last Updated May 24, 2024

Oracle Utilities Live Energy Connect Installation Guide

F74938-06

[Copyright ©](#)

Contents

- Getting Started 1
- Hardware, Software, and Networking Requirements 1
 - Hardware Requirements 1
 - Software Requirements 2
 - Networking Requirements 2
- Installing Oracle Utilities Live Energy Connect v8.0 7
 - Downloading and Running the Live Energy Connect v8.0 Installer 7
 - Creating Your Own Live Energy Connect v8.0 Installer Configuration File 9
 - Troubleshooting Installation Problems 12
 - Patching Existing Live Energy Connect v8.0 Installations 13
 - Exploring Live Energy Connect v8.0 After Installation 15
 - Verifying the Live Energy Connect v8.0 Installation 17
 - Configuring a Front-End Processor (FEP) on Your LEC v8.0 Cluster 18
 - Configuring OpenSearch Dashboard for Viewing and Searching Logs 19
 - Uninstalling Live Energy Connect v8.0 21
 - Software Included with Installation 23

Getting Started

Oracle Utilities Live Energy Connect (LEC) v8.0 is the next-generation update to the original LEC v7.1 product that is based on a microservices architecture using cloud-native tools and patterns and delivered as an on-premise solution. LEC v8.0's services run on top of a technology called Oracle Cloud Native Environment (OCNE), which is a fully integrated suite for the development and management of cloud-native applications. The LEC v8.0 installer installs OCNE v1.7 and brings up a Kubernetes (k8s) cluster on one or more Oracle Linux hosts. The installer then deploys and configures LEC V8.0's resources and services on this Kubernetes cluster. LEC V8.0's installer supports various OCNE Kubernetes cluster topologies (for example, a single-node compact cluster, a three-node compact cluster, or a cluster with three or more control nodes and two or more worker nodes).

Note: The term "node" in this document refers to a host being used in a Kubernetes cluster as a Kubernetes node. For more information about OCNE and Kubernetes terminology, refer to OCNE's "Concepts" document here: <https://docs.oracle.com/en/operating-systems/olcne/1.7/concepts/>.

The Oracle Utilities Live Energy Connect v8.0 installer is distributed as a self-extracting TAR archive that will install product components (for example, configuration files, container images, Helm charts, and so on) to the target hosts. Oracle Utilities Live Energy Connect v8.0 must be installed on a host or hosts running Oracle Linux 8 (OL8) or Oracle Linux 9 (OL9). The Oracle Utilities Live Energy Connect v8.0 installation process requires access to certain DNF repos from the Oracle Yum server and certain container images from the Oracle Container Registry. It is recommended that during installation and any patching the target host systems are configured to use a customer-managed, mirrored Yum server that can serve the required software packages and a mirrored container registry that can serve the required OCNE images. These mirrored yum servers should be provisioned from the Oracle Linux Yum server at <https://yum.oracle.com> and the Oracle Container Registry at <https://container-registry.oracle.com/v2/>. Note that where direct access to these repositories is allowed, mirroring is not required.

Hardware, Software, and Networking Requirements

The following sections outline the hardware, software, and networking requirements for using LEC v8.0.

Hardware Requirements

The minimum recommended hardware requirements for each host are:

- **CPU:** 8 CPU
- **Memory:** 16 GB RAM
- **Hard Disk Space:**
 - At least 40 GB available the `/var` directory.
 - At least 256 GB available in the `/opt` directory.
 - If you are planning to install a multinode cluster, you also need to have at least one configured block device on each host with a raw (i.e., unformatted) disk or partition that is at 256 GB in size. This will be used to back the cluster's persistent storage.
- **Architecture:** x86-64
- **Network Interface:** 1 GB Ethernet NIC
- **Filesystem:** The root filesystem should be an XFS file system (the default file system for Oracle Linux).

Software Requirements

The minimum recommended software requirements for each host are:

- **Operating System:** Oracle Linux 8 (OL8) or Oracle Linux 9 (OL9)
- **Kernel:** Unbreakable Enterprise Kernel 6 (UEK6) or Unbreakable Enterprise Kernel 7 (UEK7)
- **Security Enhanced Linux (SEL) Mode:** Permissive or Enforcing

Note: Each node in the cluster must have the same major OS release (for example, Oracle Linux 8) and the same major Kernel release (for example, UEK7). They must also have the same mode of SEL set (for example, Permissive).

Networking Requirements

Inter-node Networking Requirements (Multinode Deployment Only)

If you are installing LEC V8 in a multinode deployment, the networking and firewall rules between the nodes in your cluster need to allow for connections between certain types of nodes in the cluster on various ports. The following ports need to be open between nodes in a multinode cluster in order to deploy Kubernetes and OCNE:

The following ports need to be open between nodes in a multinode cluster in order to deploy Kubernetes and OLCNE:

- **2379/tcp:** Kubernetes etcd server client API (on control plane nodes in highly available clusters)
- **2380/tcp:** Kubernetes etcd server client API (on control plane nodes in highly available clusters)
- **6443/tcp:** Kubernetes API server (control plane nodes)
- **8090/tcp:** Platform Agent (control plane and worker nodes)
- **8091/tcp:** Platform API Server (OCNE operator)
- **8472/udp:** Flannel overlay network, VxLAN backend (control plane and worker nodes)
- **10250/tcp:** Kubernetes kubelet API server (control plane and worker nodes)
- **10251/tcp:** Kubernetes kube-scheduler (on control plane nodes in highly available clusters)
- **10252/tcp:** Kubernetes kube-controller-manager (on control plane nodes in highly available clusters)
- **10255/tcp:** Kubernetes kubelet API server for read-only access with no authentication (control plane and worker nodes)

Note: In a three node LEC v8 installation, all three nodes in the Kubernetes cluster will be control plane nodes and worker nodes. The node on which you run the installer will be the designated OCNE operator. See the following OCNE document for more information about configuring your network for a multinode OCNE cluster:

<https://docs.oracle.com/en/operating-systems/olcne/1.7/start/prereq.html#network>.

Inter-node Networking Requirements (Multinode Deployment Only)

The following LEC services running on the cluster also require certain networking considerations:

- **gRPC Config Service:** If you are using the Oracle Utilities Network Management System (NMS) Flex SCADA tool to connect to your LEC v8.0 cluster over gRPC (the typical use case for LEC v8.0), then you need to make sure that NMS can initiate a TCP connection to the LEC v8.0 cluster's IP on the specified gRPC service's port. The default port that the initial gRPC service will be listening on after a typical LEC v8.0 installation will be 50051.
- **ICCP Service:** If you will be using LEC v8.0 as an ICCP VCC that is configured to listen for inbound ICCP associations, then you need to make sure that the remote ICCP peer can initiate a TCP connection for the ICCP association to the LEC v8.0 cluster on the port that you have agreed upon with the peer. Typically, the port that an ICCP server listens on is 102.

- **keepalived Service:** If you will be installing a multinode LEC v8.0 cluster, then you can select to install keepalived on the cluster nodes in order to provide a high availability virtual IP address for the cluster. If you select this installation option then you need to make sure that Virtual Redundant Router Protocol (VRRP) and gratuitous ARP messages are allowed on the network infrastructure hosting the cluster nodes.

Note: In a single-node installation, the cluster IP from the perspective of users outside the cluster will simply be the routable IP address of the machine used as the single node in the cluster. In a multi-node installation, the cluster IP will be the high availability IP address that you specified in your LEC v8.0 configuration file.

Oracle Linux Host Configuration Requirements

Again, in order to run Oracle Utilities Live Energy Connect v8.0 installer, Oracle Linux 8 or Oracle 9 must be installed on the all nodes in the cluster. The host from which you run the LEC v8.0 installer will be designated as the OCNE operator node. In the case of a multinode deployment, this host will connect to the other hosts in the cluster via SSH while the installer is running.

If you are installing Oracle Linux from scratch on a bare metal host or hosts, you can access the Oracle Linux 8 or Oracle Linux installation images from the Oracle Linux Installation Media page. You must select a Full ISO image of the latest available version of an Oracle Linux 8 or Oracle Linux 9 release. See the Oracle Linux Help Center page for full installation instructions.

If you are installing a multinode cluster, each node in the cluster should use the same Oracle Linux major release (OL8 or OL9), have the same kernel release (UEK6 or UEK7), and have the same SEL mode enabled (Enforcing or Permissive). Once Oracle Linux 8 or Oracle Linux 9 is installed on all the nodes you plan to use for your cluster, you should run `'sudo dnf -y update'` from the terminal as a user with sudo privileges to update all of the base Oracle Linux packages and then reboot each node. Prior to proceeding with the LEC v8.0 installation, you should confirm the following prerequisites on each node:

- The user that runs the installer needs to be a non-root user with passwordless sudo privileges during the installation period.

Note: These sudo privileges are required to install OCNE packages and dependencies via DNF, to modify firewall-cmd rules, and to start certain systemd service units required for OCNE 1.7 and LEC v8.0. For more information on enabling passwordless

sudo see: https://docs.oracle.com/en/operating-systems/oracle-linux/8/userauth/userauth-GrantingsudoAccessstoUsers.html#topicudn3jt_hvb.

- The user that runs the installer needs to have passwordless SSH access to each cluster node specified in the config file during the installation period.

Note: For more information on enabling passwordless SSH see: <https://docs.oracle.com/en/operating-systems/oracle-linux/openssh/openssh-WorkingwithSSHKeyPairs.html#remote-access-without-password>

- The Bash 4 shell should be set as the default shell for the system and the user.
- The tar utility needs to be installed to extract the TAR archive from the provided .run file.
- If you are installing LEC v8.0 on OCI instances, the OCI Operating System Management Service (OSMS) must be disabled on each instance. To determine if OSMS is enabled on an instance you can run the command `ps -elf | grep osms | grep -v grep`. If a process for OSMS is returned and is running, then OSMS is enabled for that instance. Disable OSMS for the instance from the OCI management console.

Note: OSSM must be disabled on all nodes running OCNE. Unanticipated updates to Kubernetes packages and dependencies pushed through OSSM to a node can cause the Kubernetes cluster to be placed into an untenable state. All updates to OCNE components (including Kubernetes packages) should be done according to OCNE's upgrade documentation here: <https://docs.oracle.com/en/operating-systems/olcne/1.7/upgrade/#Oracle-Cloud-Native-Environment>. All other OS updates on nodes should be done according to OCNE's documentation here: <https://docs.oracle.com/en/operating-systems/olcne/1.7/upgrade/update-os.html#update-os>.

- SELinux mode must be set to Permissive or Enforcing. See the SELinux documentation for details on viewing or changing the current SELinux mode.
- The firewalld service must be running and enabled.

- Access to the Oracle Linux Yum server at <https://yum.oracle.com> (or a mirror the Oracle Linux Yum server) must be available during the installation period. The repos that the installer will require for installing OCNE 1.7 and LEC v8.0 are:
 - **For Oracle Linux 8**
 - oracle-olcne-release-el8
 - ol8_olcne17
 - ol8baseoslatest
 - ol8_appstream
 - ol8_addons
 - ol8UEK6 or ol8UEKR7
 - ol8kvmappstream
 - **For Oracle Linux 9**
 - oracle-olcne-release-el9
 - ol9_olcne17
 - ol9baseoslatest
 - ol9_appstream
 - ol9_addons
 - ol9UEK6 or ol9UEKR7

Note:The installation script will attempt to enable the correct repos from above with dnf config-manager commands. If it cannot enable these repos or install them from the DNF repos during the installation, it will exit before installing OCNE with a message that it could not enable or access a required repo. If you would like to verify your access to the yum repos and enable the required OCNE DNF repos before you run the installer to avoid any errors, refer to the following sections of OCNE's documentation:

- Oracle Linux 8 Hosts: <https://docs.oracle.com/en/operating-systems/olcne/1.7/start/prereq.html#yum-ol8>
- Oracle Linux 9 Hosts: <https://docs.oracle.com/en/operating-systems/olcne/1.7/start/prereq.html#yum-ol9>

- Access to the OCNE container images on the Oracle Container Registry at <https://container-registry.oracle.com/ords/ocr/ba/olcne> (or a mirror of that container registry):
- Any required proxies for reaching the above-mentioned resources should be defined in user's Bash environment variables: HTTP_PROXY, HTTPS_PROXY AND NO_PROXY
- If you do have HTTP_PROXY or HTTPS_PROXY defined but do not have NO_PROXY defined, then you may need to specify NOPROXY as: ",localhost,127.0.0.1," For example depending on their system and networking, a user's `.bashrc` file may need to contain something like the following snippet:

.bashrc file snippet

```
HTTP_PROXY=http://www-proxy.company.com:80/  
HTTPS_PROXY=https://www-proxy.company.com:443/  
NO_PROXY=vm-  
123,localhost,127.0.0.1,100.50.25.12,.company.corp.com,.company.vcn.com  
export HTTP_PROXY HTTPS_PROXY NO_PROXY
```

In the example `.bashrc` file snippet above, `vm-123` is the hostname of the host running the installer and `100.50.25.12` is the IPV4 address assigned to the node on the default network interface.

Installing Oracle Utilities Live Energy Connect v8.0

The following sections describe how to install and configure Oracle Utilities Live Energy Connect v8.0.

Downloading and Running the Live Energy Connect v8.0 Installer

Oracle Utilities Live Energy Connect v8.0 is distributed as a self-extracting archive. After it is downloaded to the host system, you can run the archive from a shell prompt to start the installation. Running the archive will install all components needed for Oracle Utilities Live Energy Connect v8.0.

Important: Certain steps in the installation process require root access, so the user running the installation must be a member of the `sudoers` group having `passwordless sudo`. They must have root privileges to install packages with DNF, create users (for example, `olcne`,

nginx, and so on), and configure system service units (for example, olcne-api-server.service, firewalld.service, and so on).

To install Oracle Utilities Live Energy Connect v8.0:

1. Sign into My Oracle Support.
2. Click the **Patches & Updates** tab.
3. Find the **Patch Search** section, click the **Search** tab, and then select **Product or Family (Advanced)** from the left column.
4. In the **Product** field, enter `Oracle Utilities Live Energy Connect`.
5. In the Release drop-down, select `Oracle Utilities Live Energy Connect 8.0.0.1.0`, and then click **Search**.
6. Check the **Updated** column to find the latest release and then click the Patch Name.
7. Click **Download** and then click the **TAR** link to download the archive; the archive will have a name of the form `lec_8.0.0.1.X.gz.run`.
8. Choose the host that you want to run the installer from. This will be configured as the OCNE operator during installation.
9. Move the archive to a directory from which you wish to run the installation. The directory should be owned by the user that will run the installer. As mentioned earlier, if you are creating a multi-node cluster, this user also needs to be able to SSH into each node in the cluster with password-less SSH configured.
10. From the shell, `cd` to the directory containing the archive, and enter `./<ARCHIVE>` to finish the install (where `<ARCHIVE>` is the name of the archive (for example, `lec_8.0.0.1.X.gz.run`).

Note: You may need to adjust the permissions on the `lec8.0.0.1.X.gz.run` file to run it if you encounter any errors. To do this run the following command:

```
chown ${USER}: lec_8.0.0.1.X.gz.run && chmod 700 lec_8.0.0.1.X.gz.run
```

11. The installer will then prompt if you would like to continue installing with the default settings. Press **Enter** to continue with the default installation or quit by pressing **CTRL+C**.

Note: With these default settings, a single node cluster deployment of LEC V8 will be installed on the host running the installer.

If you choose to exit out of the default installation in order to modify or create your own config file, then you can manually launch the installer afterward by running the following command:

```
./installer/install/install-lec.sh --conf <your-config-file>
```

You can use the contents of either the provided `example-conf.yaml` file or the generated `single-node-lec.yaml` file in the `install` subdirectory as a starting point for your configuration file. The following section has information about creating or modifying your LEC v8.0 installer configuration file.

Creating Your Own Live Energy Connect v8.0 Installer Configuration File

You will need to create your own **LEC v8.0 Installer Configuration** file if you do not want to use the default installer settings. You will need to create your own LEC v8.0 installation configuration file if you do not want to use the default installation settings. For example, if you want to install a multinode LEC v8.0 cluster, then you will need to create your own configuration file.

To create your own configuration file, use the included example configuration file called `/installer/install/example-conf.yaml` as a template. Refer to the configuration file parameters in the following section for information about each configuration option.

You will likely need to at least adjust the following parameters for your deployment:

- **lec-cluster.control-nodes**
- **lec-cluster.metallb-address-pool**

and for a multinode deployment:

- **lec-cluster.storage-type**
- **lec-cluster.cluster-storage-disks**

Once you've modified your configuration file, you can run the installer against it with the following command:

```
./installer/install/install-lec.sh --conf <your-config-file>
```

Note: The configuration file must contain valid YAML and be saved with a `.yaml` extension.

Live Energy Connect v8.0 Installer Configuration File Parameters

The following items are the possible parameters for LEC v8.0 installer configuration files

- **.lec-cluster.control-nodes:** List of nodes to use for the Kubernetes Control Plane. Must be a single node, three nodes, or five or more odd-numbered nodes. The nodes should be specified as either the short hostname for the host usable by the OCNE operator or an FQDN for the host useable by the OCNE operator.
- **.lec-cluster.worker-nodes:** List of nodes to use as just Kubernetes workers. Must be empty if configuring a single-node cluster or a three-node compact cluster. If specified, these nodes will only be able to be used as worker nodes in the cluster. The nodes should be specified as either the short hostname for the host usable by the OCNE operator or an FQDN for the host useable by the OCNE operator.
- **.lec-cluster.storage-type:** The storage method to use for cluster storage in the Kubernetes cluster. Needs to be one of: `rook`, `local`, or `hostPath`. This defaults to `local` for single-node clusters and “rook” for multinode clusters. If `hostPath` is specified persistence will not be available for the zookeeper, kafka, fluentd, or OpenSearch services. Using `hostPath` is not recommended for production environments. See <https://kubernetes.io/docs/concepts/storage/volumes/#hostpath>.
- **.lec-cluster.high-availability-ip:** The highly available IPV4 address used to reach the cluster. If the `use-keepalived` parameter is 'true', this will be the IPV4 that the `keepalived` services running on the cluster nodes will manage with VRRP protocol.
- **.lec-cluster.use-keepalived:** When installing a multinode cluster, you can set this parameter to "true" if you would like the LEC v8.o installer to configure keepalived services on the hosts serving as control nodes in your cluster. keepalived configures a virtual IP that can migrate between the hosts. The virtual IP that the keepalived services advertise is the IP address specified with the high-availability-ip parameter. If you plan on using a different method for providing a highly-available IP to the control node hosts in the cluster, set this parameter to false.
- **.lec-cluster.node-user:** The Linux user on each node. This user needs to be able to SSH to each node in the cluster without password prompts (using passwordless SSH with SSH keys).
- **.lec-cluster.node-ssh-key:** The path to the SSH key used to SSH from the OCNE operator to each node as the node-user user. The OCNE operator must use this same SSH key to SSH into each other node.

- **.lec-cluster.sel-mode:** The enabled SEL mode on the OCNE operator and each node in the cluster. Can be "Enforcing" or "Permissive". Cannot be "Disabled". Must be the same mode on all nodes in the cluster. If not specified, the installer will query the system to find out what the enabled SEL mode on each node is.
- **.lec-cluster.cluster-storage-disks:** The device path to raw block devices on LEC nodes that the OCNE rook module will use for storage if configuring a multinode cluster (for example, `\"/dev/sdb2\"`). Devices must be raw/unformatted block devices (disks or partitions). You can pass glob expressions (for example, `\"/dev/sdb*\"`).
- **.lec-cluster.shared-dir-size:** If configuring a multinode cluster, the size of the in-cluster shared filesystem that the OCNE Rook module will provide. Some of the LEC pods will mount this shared filesystem.
- **.lec-cluster.metallb-address-pool:** The range of IP addresses that the OCNE MetalLB module uses to advertise Kubernetes cluster services to things outside the cluster with Layer 2 advertisements. See <https://docs.oracle.com/en/operating-systems/olcne/1.7/metallb/install.html#install>. Your LEC v8.0 cluster will use MetalLB with a Layer 2 configuration. For multinode LEC V8.0 cluster deployments, the IP addresses provided for this field should be a range of reserved IPV4 addresses (e.g. "192.168.1.200-192.168.1.250") available for use on the same subnet as the nodes in the cluster. As described in the upstream MetalLB documentation at <https://metallb.universe.tf/configuration/#layer-2-configuration>, the IP addresses can be bound to the network interface that the nodes are using, "but do not necessarily need to be because MetalLB responds to ARP requests on your local network directly." In a typical LEC v8.0 installation, an nginx service will be running on each node that will route outside traffic coming into the cluster IP to these MetalLB addresses which are in turned assigned to cluster service endpoints by MetalLB. In the case of a single-node LEC v8.0 cluster, the range of IP addresses specified is more flexible because traffic routed from nginx to the IP addresses that MetalLB assigns does not leave the node. The range of addresses just needs to not conflict with any existing routes configured on the node before installing LEC v8.0.

Example Live Energy Connect v8.0 Installer Configuration Files

The following content is in example of an LEC v8.0 installer configuration file for a single node cluster that specifies using the `hostPathstorage` mode:

Sample LEC v8.0 Single Node Installer Configuration File

```
lec-cluster:
  # K8S & OCNE Info
  control-nodes:
    - node2
  worker-nodes:
```

```
storage-type: hostPath
# Node Config Info
node-user: opc
node-ssh-key: /home/opc/.ssh/id_rsa
sel-mode: Enforcing
# LEC Apps Info
metallb-address-pool: 192.168.1.200-192.168.1.250
```

The following content is an example of an LEC v8.0 installer configuration file for a three-node compact cluster that specifies using the OCNE Rook module for storage:

Sample LEC v8.0 Three Node Installer Configuration File

```
lec-cluster:
# K8S & OCNE Info
control-nodes:
- node1
- node2
- node3
worker-nodes:
storage-type: rook
high-availability-ip: 192.168.122.111
use-keepalived: true
# Node Config Info
node-user: opc
node-ssh-key: ~/.ssh/id_rsa
sel-mode: Enforcing
# LEC Apps Info
cluster-storage-disks: /dev/vdb
shared-dir-size: 50
metallb-address-pool: 192.168.1.200-192.168.1.250
```

Troubleshooting Installation Problems

The following is a list of troubleshooting steps you can use if you encounter certain problems during the installation process:

Problem: Uncompressing Oracle Utilities Live Energy Connect ... Extraction failed

When the self-extracting archive is run, it will use the tar program to untar the contents of the installer. If it cannot call the tar command it will print the following error:

```
[opc@nod1 ~]$ ./lec 1.0.0-987836d.gz.run Creating directory installer
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing Oracle Utilities Live Energy Connect ... Extraction
failed. Terminated [opc@nod1 ~]$ 0% ... Decompression failed.
```

Solution: If this happens, you can create the users ahead of time by running the following commands before re-running the installer script:

```
sudo dnf install tar
```

Problem: Cannot create user "olcne"

When the olcne-utils DNF package is installed, it will attempt to create a user called "opc" and a group called "olcne". If it is not able to do this, the installer will exit.

Solution: If this happens, you can create the users ahead of time by running the following commands before re-running the installer script:

```
sudo groupadd -r olcne
```

```
sudo useradd -r -s /usr/sbin/nologin -g olcne -d <user_home_path> -m
-c "User created to manage olcne privileged commands" olcne
```

If you still cannot create a user using the above steps, contact your Linux system administrator. They may need to modify the system settings (for example, `logins.def`) for automatically adding users and groups. For more information see

https://docs.oracle.com/en/operating-systems/oracle-linux/8/userauth/userauth-WorkingWithUserandGroupAccounts.html#topic_qnx_hdx_1tb

Patching Existing Live Energy Connect v8.0 Installations

About Patching LEC v8.0

To patch the LEC v8.0 applications and resources running on your LEC v8.0 cluster, you need to run the `patch-lec.sh` script included in the patch's installer payload.

To upgrade OCNE from OCNE 1.7 to 1.8, you need to follow the OCNE update and upgrade instructions in the documentation at <https://docs.oracle.com/en/operating-systems/olcne/1.8/upgrade/upgrade.html>.

To patch your Oracle Linux system, including running any DNF updates, you need to follow the OCNE instructions at <https://docs.oracle.com/en/operating-systems/olcne/1.8/upgrade/update-os.html#update-os>.

Running the LEC v8.0 Patching Script

To patch your LEC v8.0 applications if upgrading an existing LEC v8.0 installation you need to run the `patch-lec.sh` script included in the patch's installer payload.

Important: Certain steps in the patching process require root access, so the user running the installation must be configured in the sudoers configuration appropriately with passwordless sudo. They must have root privileges to move container image, chart, and LEC script files and to push images to the podman cache with podman.

To patch Oracle Utilities Live Energy Connect v8.0:

1. Sign into My Oracle Support.
2. Click the **Patches & Updates** tab.
3. Find the **Patch Search** section, click the **Search** tab, and then select **Product or Family (Advanced)** from the left column.
4. In the **Product** field, enter Oracle Utilities Live Energy Connect.
5. In the **Release** drop-down, select **Oracle Utilities Live Energy Connect 8.0.0.0.0**, and then click **Search**.
6. Check the Updated column to find the latest release and then click the **Patch Name**.
7. Click **Download** and then click the TAR link to download the archive. The archive will have a name of the form `lec_8.0.0.1.X.gz.run`.
8. Choose the host from which you want to run the patching script. This should be the same machine from which you originally installed Live Energy Connect.
9. Move the archive to a directory from which you wish to run the installer. The directory should be owned by the user that will run the installer. As mentioned earlier, if you are patching a multinode cluster, this user also needs to be able to SSH into each node in the cluster with passwordless SSH configured.
10. From the shell, `cd` to the directory containing the archive and Enter `./lec_8.0.0.1.X.gz.run` to run the self-extracting archive file.

Note: You may need to adjust the permissions on the `lec_8.0.0.1.X.gz.run` file to run it if you encounter any errors. To do this run the following command:

```
chown ${USER}: lec_8.0.0.1.X.gz.run && chmod 700 lec_8.0.0.1.X.gz.run
```

11. After running the self-extracting archive, you will then be prompted if you would like to continue installing with the default settings. When prompted press **CTRL + C** to quit the prompt. To then run the patching script, run:

```
./installer_8.0.0.1.x/install/patch-lec.sh --lec-env-file  
~/lec/lec.env
```

12. When prompted by the patch script, press **Enter** to continue with the patching.
13. If the patch script fails to patch the Live Energy Connect v8, it will then attempt to revert back to the original installation. If you encounter problems in the patching process contact My Oracle Support.

Exploring Live Energy Connect v8.0 After Installation

About the LEC v8.0 Kubernetes Cluster

After the LEC v8.0 installation is complete LEC v8.0 application will be running on an OCNE Kubernetes cluster.

kubectl

`kubectl` is a CLI tool for communicating with a Kubernetes cluster's Control Plane API's. You can use the `kubectl` utility to explore your OCNE Kubernetes cluster and LEC v8.0 deployments. `kubectl`'s upstream reference documentation is located here:

<https://kubernetes.io/docs/reference/kubectl/>. The installer creates a `kubectl` configuration file for the Linux user running the LEC v8.0 installer at `~/.kube/config`. This kube config file will allow the Linux user to access the cluster as an administrator (as the default user-facing clusterRole called "cluster-admin") with the `kubectl` CLI. You can learn more about the default cluster-admin role here: <https://kubernetes.io/docs/reference/access-authn-authz/rbac/#user-facing-roles>. At install time, this same cluster-admin configuration file will also be placed in the kubernetes system files owned by root at `/etc/kubernetes/admin.conf` on each control node in the cluster. During the LEC v8.0's deployment, the installer creates a Kubernetes namespace called "lec" into which it deploys the LEC-specific cluster resources on your OCNE Kubernetes cluster. For convenience, the installer will add a `kubectl` context to your kube config file called `lec-admin` so that `kubectl` CLI commands will specify this namespace by default. If you wish to use `kubectl` to interact with other cluster resources in different namespaces (for example, "default", "system", "rook", and so on) either specify the namespace in your `kubectl` command or use the "-A" option. You can learn more about using `kubectl` to explore your LEC v8.0 Kubernetes cluster by looking at OCNE's `kubectl` tutorial here: <https://docs.oracle.com/en/learn/ocne-kubectl-intro/index.html#view-context-and-configuration-information>.

Environment Variables

The LEC v8.0 installer will define a number of environment variables in a file at `~/ .lec/lec.env`. A line is also added to the Linux user's `.bashrc` profile to load the `~/ .lec/lec.env` file at login. These environment variables make it easier to monitor and troubleshoot your OLCNE Kubernetes cluster and LEC v8.0 installation. The values for the environment variables in the `~/ .lec/lec` file are determined by the installer based on your system and the configuration file you used with the installer. You do not need to update the values of these environment variables manually.

The following list contains example values for the environment variables that get defined in the `~/ .lec/lec` file. Some of the values in your `~/ .lec/lec` file will look different.

Example Environment Variable Definitions

```
KUBECONFIG=/home/opc/.kube/config
LEC_DIR=/opt/lec
LEC_BUILD_VERSION=8.0.0.1.2.33-6bad016
LEC_OLCNE_PREFIX=lec
LEC_OPERATOR_NODE=node1
LEC_COMMON_DIR=/opt/lec/common
LEC_NODES_NETWORK_INTERFACE=eth0
LEC_NAMESPACE=lec
OLCNE_REGISTRY=container-registry.oracle.com/olcne
LEC_ROOK_NODES=node1 node2 node3
LEC_INITIAL_INSTALL_TIME=270324T044250
LEC_PODMAN_STORAGE_DIR=/opt/lec/podman_storage
LEC_REGISTRY=localhost
LEC_ADMIN_CONTEXT=lec-admin
LEC_CLUSTER_TYPE=multiple
LEC_ENV_DIR=/home/opc/.lec
LEC_INSTALL_LOGS=/var/log/lec/install
LEC_CONF_BACKUPS_DIR=/opt/lec/backups
LEC_HIGH_AVAIL_IP=192.168.122.111
LEC_USER=opc
LEC_ALL_NODES=node1 node2 node3
LEC_USE_KEEPALIVED=true
LEC_SSH_KEY=/home/opc/.ssh/id_rsa
LEC_ROOK_NAMESPACE=rook
LEC_ROOK_STORAGE_DISK_DEVICE=/dev/vdb
LEC_VERSION=8.0.0.1.2
LEC_INITIAL_BUILD_VERSION=8.0.0.1.2.33-6bad016
OLCNE_VERSION=olcne17
LEC_LAST_PATCH_TIME=
```

```
LEC_STORAGE_TYPE=rook
LEC_CONTROL_NODES=node1 node2 node3
OLCNE_INITIAL_VERSION=olcne17
LEC_WORKER_NODES=
```

Using kubectl to Explore your LEC v8.0 Applications and OCNE Kubernetes Cluster

Oracle Utilities Live Energy Connect v8.0 runs as a collection of microservices on an Oracle Cloud Native Environment (OCNE) Kubernetes cluster. You can use the kubectl client CLI to explore the cluster after it is installed.

Verifying the Live Energy Connect v8.0 Installation

After you have installed Oracle Utilities Live Energy Connect v8.0, you can verify that the LEC v8.0 resources are deployed with the **kubectl client** CLI tool. To do this, try the following steps:

1. Reload the user's Bash environment variables by exiting the interactive shell and then starting a new one.

Alternatively, you could run the command:

```
source ~/.bashrc
```

2. At the command prompt check the status of all the pods in the "lec" namespace of the Kubernetes cluster by running the command:

```
kubectl get pods
```

This command should return a list of the LEC-specific pods in the cluster with information about their running status. If the installation was successful the status should say `Running`. For example, the output of that command should look something like:

```
[opc@node1 ~]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
fluentd-fld-fluentd-2g5qr          1/1     Running   1           4m
fluentd-fld-fluentd-knzfb          1/1     Running   1           4m
fluentd-fld-fluentd-m4fm7          1/1     Running   1           4m
kafka-0                             1/1     Running   4 (58m ago) 3m
kafka-zookeeper-0                  1/1     Running   1           3m
opensearch-cluster-master-0        1/1     Running   1           5m
os-dashboards-lec-747cdcd4cf-8874d 1/1     Running   1           5m
[opc@node1 ~]$
```

3. If you want to get more information about a particular pod you could use the `kubectl describe pod` command:

```
[opc@node1 ~]$ kubectl describe pod kafka-0
Name:          kafka-0
Namespace:     lec
Priority:       0
Service Account: kafka
Node:          node1/192.168.122.101
Start Time:    Wed, 27 Mar 2024 04:45:24 +0000
Labels:        app.kubernetes.io/component=kafka
               app.kubernetes.io/instance=kafka
               app.kubernetes.io/managed-by=Helm
               app.kubernetes.io/name=kafka
               controller-revision-hash=kafka-57445667b
               helm.sh/chart=kafka-20.0.2
               statefulset.kubernetes.io/pod-name=kafka-0
Annotations:   <none>
Status:        Running
IP:            10.244.0.20
IPs:
IP:            10.244.0.20
Controlled By: StatefulSet/kafka
...
```

Note: You can use `kubectl` to list and describe other cluster resources not just Pods (for example, Services, Deployments, PV's, PVC's, and so on). Refer to the upstream Kubernetes `kubectl` documentation at <https://kubernetes.io/docs/reference/kubectl/> to learn more about using `kubectl` to inspect your cluster.

Configuring a Front-End Processor (FEP) on Your LEC v8.0 Cluster

LEC v8.0's message bus configuration is driven by configurations defined by clients (for example, NMS FlexSCADA) which send configuration information to LEC v8.0. From the client's perspective, LEC v8.0 acts as a Front-End Processor (FEP) for the client. **Note:** A single LEC v8.0 cluster can host multiple FEPs. For example, NMS Flex-SCADA users will use the LEC v8.0 cluster's various applications by creating a FEP in NMS and configuring it to connect to a particular FEP service running on the LECv8.0 cluster. The FEP service (for example, "fep7-grpc-service") running on the LEC v8.0 cluster will then create and manage the resources requested by NMS FlexSCADA. By default, no FEP services are deployed on your LEC v8.0 cluster. The

FEP service needs to be initialized with a specified, unique FEP ID that may vary depending on the client's configuration state.

You can use the following steps to create and configure a FEP on your LEV v8.0 cluster so that a client like NMS flexSCADA can use the FEP service. These steps will install the FEP service on the cluster and configure the required nginx service configuration and firewalld configuration on the nodes in the cluster.

Add a FEP Service on Your LEC v8.0 Cluster

On the designated OCNE operator (i.e. the node from which you initially ran the LEC v8.0 installer) do the following:

1. Open a Bash shell session and cd to the `/opt/lec/scripts` directory.
2. Run the following command:

```
./create-fep.sh --lec-env-file ~/.lec/lec.env --id <fep_id> --port <port_number>
```

where is the unique numeric ID that you want the FEP service to have and is the port on which you want the FEP service to listen. For example, you may run something like:

```
./create-fep.sh --lec-env-file ~/.lec/lec.env --id 7 --port 50051
```

3. You will be prompted to hit Enter to continue with this step. You can also exit using **CTRL + C**.
4. When the script finishes running the FEP service "fepgrpc-service" (for example, "fep7-grpc-service") will no longer be deployed on the cluster. Note: It may take a few minutes for the FEP and its related cluster resources to be completely uninstalled depending on the number and type of resources that need to be uninstalled. You can confirm that the FEP service was removed by looking at all the services in the cluster's "lec" namespace after it was removed. Again, you can do this from the CLI using kubectl. For example:

```
kubectl get services
```

Configuring OpenSearch Dashboard for Viewing and Searching Logs

By default, the OpenSearch Dashboard application is deployed on your LEC v8.0 cluster after installation. However, access to the OpenSearch Dashboard via HTTP is not configured during your LEC v8.0 installation. In other words, the OpenSearch Dashboard's application resources are deployed on the cluster during the LEC v8.0 installation but the nginx service is not configured and the firewalld rules for this endpoint are not added during the installation. You can use the following steps to configure your host (or host's) nginx services so that the OpenSearch Dashboard web application is reachable via HTTP. Since the OpenSearch Dashboard web app is served with HTTP, you should only allow trusted hosts to connect to the service endpoint.

Configure NGINX Service for Proxying OpenSearch Dashboard HTTP Connections

On the designated OCNE operator (i.e. the node from which you initially ran the LEC v8.0 installer) do the following:

1. Open a Bash shell session and cd to the `/opt/lec/scripts` directory. Run the following command:

```
./configure-osdb-access.sh --lec-env-file ~/.lec/lec.env --mode HTTP
```

Note: The typical port that OpenSearch Dashboards web applications use is 5601 but you can specify another with the "--port" parameter.

2. You will be prompted to hit Enter to continue with this step. You can also exit using **CTRL + C**.
3. When the script finishes running the OpenSearch Dashboard web application should be served via HTTP on port 5601 of the external cluster IP. In a single node deployment this IP address will be the routable IP address of the single node in the cluster. In a multinode deployment this will be the routable, highly available (HA) virtual IP address of the cluster.
4. Using a browser that can reach your LEC v8.0 cluster, browse to `http://:5601` where is the IP address or FQDN of the cluster. You should see the Opensearch Dashboards landing page.

In order to start using the Opensearch Dashboard application to view and search your LEC v8.0 application logs, you need to create an index pattern for the LEC log files that OpenSearch has collected.

Create an OpenSearch Index Pattern on OpenSearch Dashboard

1. Browse to `http://:5601` where is the IP address or FQDN of the cluster.
2. When prompted with the initial splash screen, click on Explore on my own.
3. Click the dropdown menu icon in the top left, and select the Stack Management option from the dropdown menu.
4. On the screen that opens, click the option called Index Patterns listed at the upper left.
5. A menu called "Create index pattern" will be displayed. In the field called Index pattern name enter the string "logstash-". This will create an index pattern that matches the LEC OpenSearch indices created by OpenSearch. Note: If you do not want to create one index pattern for all of your LEC v8.0 OpenSearch logs, then you can specify a more specific index pattern in this step. The indices created for LEC logs by OpenSearch are organized and

named by date (for example, “logstash-2024.03.12”, “logstash- 2024.03.13”, .etc.).

6. When prompted click the Next step button.
7. When prompted to choose a timestamp field, select the "@timestamp" field. Then when prompted click the Create index pattern button.
8. Now that the index pattern is created, click the dropdown menu icon in the top left again and now select the Discover option from the dropdown menu.
9. Logs matching the index pattern will be displayed in the OpenSearch Dashboard’s Discover page.

Note: You can use the Search bar on the Discover page to search for specific logs using the Dashboards Query Language (DQL). For more information on using DQL, refer to the upstream OpenSearch Dashboard documentation here:

<https://opensearch.org/docs/latest/dashboards/dql/>

Uninstalling Live Energy Connect v8.0

A number of scripts are provided that can be used to uninstall all of an LEC v8.0 deployment or only certain subcomponents. After installation these scripts live in the directory at /opt/lec/scripts. Running any of the uninstall scripts will permanently change your LEC v8.0 deployment. Use them with caution. Each uninstall script requires a parameter called --lec-env-file file to run. The value of this parameter should be the absolute filepath to a shell-style file that defines the LEC-specific environment variables that are required to uninstall LEC v8.0. Users do not need to create this file. During the installation one was created for the user at ~/.lec/lec.env. The uninstall scripts need to be run from the same host that the installation script was run (i.e. the designated OCNE operator) and by the same user. And like the installation, the user running the uninstallation scripts will need to have passwordless SSH access to all the nodes in the cluster and be able to use passwordless sudo on all the nodes in the cluster during the uninstallation period. The following is a list of the provided uninstall scripts and a description of what they remove.

- **remove-lec-apps . sh:** This script will attempt to uninstall the LEC v8.0 specific cluster resources running on your OCNE Kubernetes cluster.
- **remove-rook . sh:** This script will attempt to uninstall the OCNE Rook module and its related cluster resources in your multinode LEC v8.0. In order to run this script, you must have already uninstalled your LEC v8.0 cluster resources with the scripts above. Note that a single-node LEC v8.0 cluster will not have the OCNE Rook module installed.

- **remove-olcne-modules . sh:** This script will attempt to uninstall the OCNE modules and their related services that were used to provide your LEC v8.0 cluster. In order to run this script, you must have already uninstalled your LEC v8.0 cluster resources with the scripts above.
- **remove-podman . sh:** This script will attempt to remove and reset the podman package on all of the nodes in the LEC v8.0 cluster and remove any orphaned podman storage space. In order for this script to remove podman, all LEC applications and all OCNE modules and services must have already been uninstalled with the above scripts.
- **remove-olcne-packages . sh:** This script will attempt to uninstall the OCNE-related packages that were installed on the nodes in the LEC v8.0 cluster during the initial installation. In order to run this script, you must have already uninstalled the LEC V8 cluster resources, the OCNE modules, and the podman package with the above scripts.
- **remove-lec-nginx-config . sh:** This script will attempt to disable the nginx services that were installed on the nodes in the LEC v8.0 cluster depending on the initial specified LEC v8.0 installation.
- **remove-lec-keepalived-config . sh:** This script will attempt to disable the keepalived services that may have been installed on the nodes in the LEC v8 cluster depending on the initial specified LEC v8.0 installation. Note that single-node LEC v8.0 clusters will not have keepalived installed.
- **remove-lec-firewalld-rules . sh:** This script will attempt to remove the firewalld rules for LEC v8.0 application that were created on the nodes in the LEC v8.0 cluster during installation.
- **bash-utils . sh:** This is just a helper script that defines common functions/commands used by the other scripts in this directory.
- **remove-all-lec . sh:** This script will run all of the other scripts above in the correct order (as necessary) and in doing so attempt to remove the entirety of the install footprint of the LEC v8.0 cluster. See the next section on how to run this script.

Note: Each uninstall script in this directory can display a usage message specifying its available parameters. To see this message run: `<script> --help`

Uninstall All of the Live Energy Connect v8.0 Deployment

To uninstall all of the LEC deployment v8.0, cd to the directory at `/opt/lec/scripts` and run the following command:

```
./remove-all-lec.sh --lec-env-file ~/.lec/lec.env --backup-fep-configs  
--remove-packages --remove-files
```

At each step in the uninstall process, you will be prompted to hit Enter to continue with that step. You can exit using **CTRL + C**. **Note:** After LEC is uninstalled a back up of the FEP configuration information your LEC v8.0 deployment and a backup of your `~/.lec/lec.env` file will have been stored as zip files in the directory at `/opt/lec/backups`.

Software Included with Installation

The LEC installer will install the following software on the nodes in your cluster. All open source software licensing information can be found in the Oracle Utilities Live Energy Connect Licensing Information User Manual.

- Oracle OpenSearch
- Oracle OpenSearch Dashboard (used for viewing logs)
- Oracle Cloud Native Environment