Oracle® Communications Diameter Signaling Router

Oracle Certificate Lifecycle Manager Guide

Release 9.3.1.0.0
G58416-01
August 2026

Oracle Communications Diameter Signaling Router Oracle Certificate Lifecycle Manager Guide, Release 9.3.1.0.0

G58416-01

Contents

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
    - For Technical issues such as creating a new Service Request (SR), select **1**.
    - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

What's New in This Guide

This section introduces the documentation updates for Release 9.3.1.0.0.

Release 9.3.1.0.0 - G58416-01 , August 2026

First publication of the OCLM (Oracle Certificate Lifecycle Manager) guide.

Acronyms

The following table provides information about the acronyms used in this document.

Table Acronyms

Acronym	Meaning
OCLM	Oracle Communications Lifecycle Management
CMP	Certificate Management Protocol
NOAM	Network Operations, Administration, and Maintenance
SOAM	System Operations, Administration, and Maintenance
HA	High Availability
CA	Certificate Authority
CSR	Certificate Signing Request
DN	Distinguished Name
DSR	Diameter Signaling Router
SDS	Subscriber Data Server
REST	Representational State Transfer
API	Application Programming Interface
TLS	Transport Layer Security
HTTPS	Hypertext Transfer Protocol Secure
IR	Initial Registration
KUR	Key Update Request
VM	Virtual Machine
VMDK	Virtual Machine Disk
QCOW2	QEMU Copy On Write version 2
PKI	Public Key Infrastructure
GUI	Graphical User Interface
SAN	Subject Alternative Name
IP	Internet Protocol
DNS	Domain Name System

1 Introduction

Oracle Communications Lifecycle Management (OCLM) provides a VM-based certificate lifecycle service for requesting and renewing certificates from an Operator CA. OCLM is used in the DSD/SDS certificate lifecycle environment and must be configured with the customer DSD/SDS deployment and Operator CA.

This guide describes the OCLM node layout, Operator CA CMP configuration, Certificate REST APIs, Deployment operations, Upgrade, Patch procedures, Active-Active OCLM Deployment, and Troubleshooting. This guide enables operations teams to install and operate OCLM in a customer environment.

1.1 References

The following table provides customer-specific references while installing or operating OCLM:

Table 1-1 References

Reference	Purpose
OCLM VM/VMDK delivery package	Provides the approved OCLM VM image and release-specific artifacts.
Customer DSD/SDS deployment details	Provides the DSD/SDS integration context, HA endpoint configuration, and approved client connectivity.
Operator CA deployment details	Provides CMP endpoint, recipient DN, issuer DN, trust files, and TLS requirements.

> **Note**
>
> The OCLM delivery does not include a CSAR package. OCLM is delivered as an approved VMDK or QCOW2 image.

1.2 Terminology

The following table provides information about the terminology used in this document.

Table 1-2 Terminology

Acronym	Meaning
OCLM	The VM based certificate lifecycle service described in this guide.
DSD/SDS	Customer deployment environment that integrates with OCLM for certificate lifecycle operations.

Table 1-2 (Cont.) Terminology

Acronym	Meaning
Operator CA	Certificate Authority used by OCLM for certificate issuance and renewal through CMP.
CMP	Certificate Management Protocol used between OCLM and Operator CA.
CMP identity	The OCLM client certificate and private key used to sign CMP requests to Operator CA.
NOAM	DSR/SDS Network Operations, Administration, and Maintenance node used for central management functions.
SOAM	DSR/SDS System Operations, Administration, and Maintenance node used for site-level management functions.
HA	High Availability. In this guide, HA refers to DSR/SDS using two active OCLM nodes configured independently.

2 Overview

OCLM receives certificate requests over HTTPS, validates request payloads, requests certificate issuance or renewal from Operator CA, and returns issued certificate material to the caller.

The service uses local files under `/opt/oclm-service` for runtime issuer configuration and CMP identity for signature-based trust establishment with Operator CA.

2.1 Certificate Create

A client submits a certificate request to `POST /oclm-config/v1/certs`. OCLM requests certificate issuance from Operator CA, and the client retrieves the issued certificate material with `GET /oclm-config/v1/certs/{name-or-uuid}`.

2.2 Certificate Retrieve

A client uses `GET /oclm-config/v1/certs/{name-or-uuid}` to retrieve certificate material for a completed certificate create request. After successful retrieval, OCLM returns the generated certificate and private key material to the caller and removes the temporary generated files for that request identifier.

2.3 Certificate Renew

A client submits the existing certificate and renewal request to `POST /oclm-config/v1/certs/{name}/renew`. OCLM requests renewal from Operator CA and returns the renewed certificate material in the API response.

2.4 Active-Active OCLM Deployment

DSR/SDS supports an Active-Active OCLM deployment by configuring two active OCLM nodes in the DSR/SDS GUI. In this model, both OCLM nodes are available for certificate lifecycle requests, and both OCLM nodes can communicate with Operator CA.

The two OCLM nodes are deployed and managed as independent active nodes. DSR/SDS is responsible for using the configured OCLM nodes. OCLM does not provide node-to-node clustering or automatic synchronization between the nodes.

For an Active-Active OCLM pair, ensure the following for both OCLM nodes:

- The OCLM VM is deployed and reachable from DSR/SDS on TCP 8989.
- Operator CA CMP configuration is completed in `/opt/oclm-service/issuerConfig.properties`.
- Operator CA trust files are installed in the `/opt/oclm-service`.
- Initial CMP client certificate and key are installed in the `/opt/oclm-service/certs`.
- The OCLM service is running and listening on TCP 8989.

- OCLM can reach the Operator CA CMP endpoint.
- Operator CA trusts the CMP client identity used by the OCLM node according to the customer PKI policy.

3 Prerequisites

Customers deploy OCLM from a supplied VMDK image. The image contains the OCLM service payload, service launcher scripts, `systemd` service configuration, log configuration, and runtime directory structure.

The OCLM service should be started only after Operator CA configuration, initial CMP identity files, and required trust files are installed.

Before starting, patching, or upgrading the OCLM node, collect the following site-specific inputs from the virtualization, networking, PKI, and DSD/SDS teams.

Table 3-1 Prerequisites

Input	Required Information
VM image	Approved OCLM image artifact
Administrative access	Initial VM login method and approved password rotation procedure.
Operator CA endpoint	CMP server URL, port, path, recipient DN, and timeout policy
Operator CA trust	Root and intermediate CA certificate files used to verify issued certificates.
Operator CA TLS trust	TLS issuing CA for the Operator CA HTTPS endpoint, if TLS validation is enabled.
Initial OCLM identity	Initial CMP client certificate and matching private key for signature-protected CMP requests.

3.1 VM Requirements

The following table provides the VM requirements:

Table 3-2 VM Requirements

Resource	Default/Minimum
vCPU	2
Memory	2048 MB
Disk	40 GB or larger
Network	One routable IPv4 or IPv6 network interface

The following OCLM service settings are preconfigured in the supplied OCLM image:

Table 3-3 Preconfigured OCLM Service Details

Resource	Preconfigured value
OCLM application port	TCP 8989

Table 3-3 (Cont.) Preconfigured OCLM Service Details

Resource	Preconfigured value
OS user for OCLM service	admusr
Service home	/opt/oclm-service
Main log	/var/log/oclm.log

> **Note**
>
> Ensure that the external network and security group rules permit DSR/SDS clients to access the OCLM service on this port.

3.2 Network Requirements

The OCLM VM requires customer approved network connectivity for the following:

- DSR/SDS clients must be able to reach OCLM on TCP 8989.
- OCLM must be able to reach the Operator CA CMP endpoint.

3.3 Operator CA Requirements

Operator CA must provide the CMP endpoint, recipient DN, issuer DN if required by policy, issued-certificate trust chain, and TLS trust details when CMP TLS validation is enabled.

Operator CA must trust the OCLM CMP client identity used for signature-protected CMP requests.

3.4 DSR/SDS Integration Requirements

DSR/SDS must be configured with the OCLM endpoint details that it will call for certificate lifecycle operations. For HA deployments, configure both active OCLM endpoints in the DSR or SDS GUI.

Required integration inputs include:

- OCLM Node 1 IP and HTTPS port 8989.
- OCLM Node 2 IP and HTTPS port 8989, when Active-Active HA is used.

4

Installation and Configuration

4.1 Upload OCLM VMDK to OpenStack

The Oracle-provided OCLM VMDK must be uploaded to the customer OpenStack cloud image service before an OCLM instance can be created.

> **Note**
>
> This command can change based on the customer cloud environment, image visibility policy, project or tenant configuration, and approved operational procedure.

```
openstack image create --private \
  --disk-format vmdk \
  --container-format bare \
  --file /path/to/oclm-<version>.vmdk \
  "oclm-vmdk-<version>.vmdk"
```

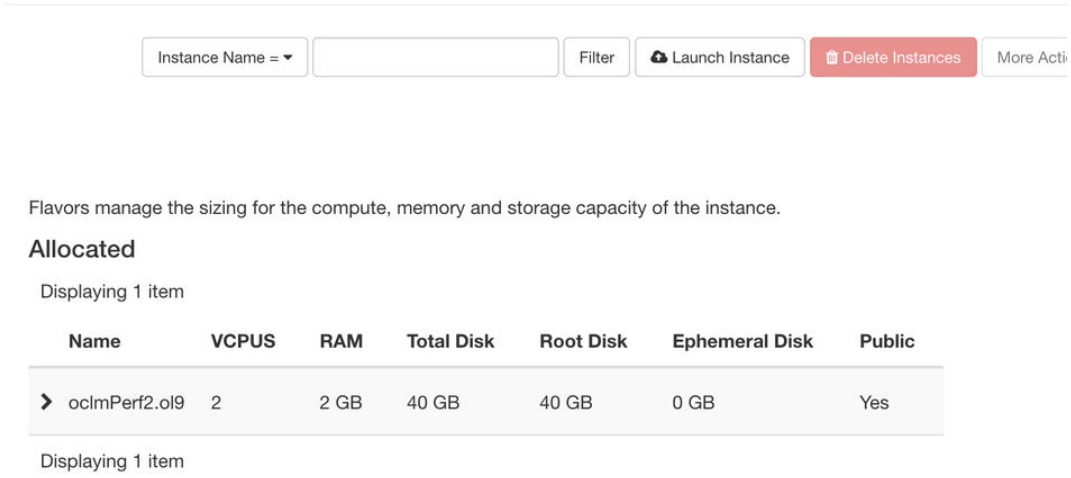
Validate that the image is available in OpenStack:

```
openstack image show "oclm-vmdk-<version>.vmdk"
```

4.2 Create the OCLM Instance

Create an OCLM instance from the uploaded OCLM image using the customer-approved OpenStack instance creation procedure. Select a flavor that meets the minimum VM requirements, network, security group and IPv4 or IPv6 addressing.

Figure 4-1 OCLM Instance



> **Note**
>
> Select the flavor with minimum OCLM VM requirements.

4.3 Access the OCLM Instance

After the VM is active and networking is assigned, log in from an approved administration host:

```
ssh admusr@<oclm-node-ip>
```

Change the password at the first login by using the default password:

```
sudo passwd admusr
```

4.4 Install Operator CA Trust Files

Copy the required Operator CA trust certificates to the OCLM node by using a customer-approved secure transfer method. Stage the files in `/home/admusr/oclm-staging/operator-ca-trust`, install them into `/opt/oclm-service`, and remove the staged copies after successful installation.

The following table provides trust files used by Operator CA configuration:

Table 4-1 Trust Files

File	Required	When to Install	Configuration Property
/opt/oclm-service/operator-ca-root.pem	Mandatory	Always required. This is the trust anchor used to verify certificates issued by Operator CA.	cmp.rootCACerts
/opt/oclm-service/operator-ca-intermediate.pem	Conditional	Install only when Operator CA issues certificates from an intermediate CA or when the issued certificate chain cannot be verified with the root CA alone. If only the root CA is used, this file is not required.	cmp.rootCACerts
/opt/oclm-service/operator-ca-tls-root.pem	Conditional	Install only when the CMP endpoint uses HTTPS and `cmp.enableTLS=true`. If `cmp.enableTLS=false`, or the CMP endpoint does not require TLS validation, this file is not required.	cmp.tlsTrustedCerts

`cmp.rootCACerts` must include the files needed to verify certificates issued by Operator CA. If the issuing CA is the root CA, configure only the root file. If Operator CA uses an intermediate CA, configure both the intermediate and root files as a Comma-separated list. The paths in `issuerConfig.properties` must exactly match the installed file paths.

Example with only a root CA:

```
cmp.rootCACerts=/opt/oclm-service/operator-ca-root.pem
```

Example with an intermediate CA:

```
cmp.rootCACerts=/opt/oclm-service/operator-ca-intermediate.pem,/opt/oclm-
service/operator-ca-root.pem
```

Example when CMP TLS validation is disabled:

```
cmp.enableTLS=false
# cmp.tlsTrustedCerts=/opt/oclm-service/operator-ca-tls-root.pem
```

Example when CMP TLS validation is enabled:

```
cmp.enableTLS=true
cmp.tlsTrustedCerts=/opt/oclm-service/operator-ca-tls-root.pem
```

Run the following commands to create the staging directory:

```
STAGE_DIR=/home/admusr/oclm-staging/operator-ca-trust
mkdir -p "$STAGE_DIR"
chmod 700 "$STAGE_DIR"
```

Copy only the required Operator CA trust files into $STAGE_DIR using the customer-approved secure transfer method. Then install the files from the staging directory:

```
sudo install -m 0644 "$STAGE_DIR/operator-ca-root.pem" /opt/oclm-service/
operator-ca-root.pem
[ -f "$STAGE_DIR/operator-ca-intermediate.pem" ] && sudo install -m 0644
"$STAGE_DIR/operator-ca-intermediate.pem" /opt/oclm-service/operator-ca-
intermediate.pem
[ -f "$STAGE_DIR/operator-ca-tls-root.pem" ] && sudo install -m 0644
"$STAGE_DIR/operator-ca-tls-root.pem" /opt/oclm-service/operator-ca-tls-
root.pem
sudo chown admusr:admusr /opt/oclm-service/operator-ca-root.pem
[ -f /opt/oclm-service/operator-ca-intermediate.pem ] && sudo chown
admusr:admusr /opt/oclm-service/operator-ca-intermediate.pem
[ -f /opt/oclm-service/operator-ca-tls-root.pem ] && sudo chown
admusr:admusr /opt/oclm-service/operator-ca-tls-root.pem
```

Perform the following command to remove the staged copies after the files are installed:

```
rm -f "$STAGE_DIR"/operator-ca*.pem
rmdir "$STAGE_DIR" 2>/dev/null || true
```

4.5 Configure Operator CA

The Operator CA details identify the CA authority used by OCLM. Configure these details in:

```
/opt/oclm-service/issuerConfig.properties
```

This file contains the Operator CA CMP server details, CA certificate references, TLS trust configuration, recipient DN, issuer DN, and timeout settings.

Use the provided configuration script to update `issuerConfig.properties`. The script prompts for each CMP property, shows a short definition and sample value, reuses existing values as defaults, and updates the file in place after confirmation.

Run the following script on the OCLM node:

```
cd /opt/oclm-service
python3 configureIssuer.py
```

The following table provides required properties:

Table 4-2 Required Properties

Property	Description	Sample Value
cmp.recipientDN	Recipient Distinguished Name expected by Operator CA for CMP messages.	/C=SE/O=EJBCA Sample/CN=ManagementCA
cmp.rootCACerts	Comma separated list of local CA certificate files used to verify certificates issued by Operator CA. At least the Operator CA root certificate is required.	/opt/oclm-service/operator-ca-root.pem
cmp.server	Operator CA CMP endpoint URL. Include scheme, host, port, and any required CMP alias/path. Mask IP addresses in shared documentation.	http://10.xxx.xxx.xxx:8080/ejbca/publicweb/cmp/oclm

Table 4-3 Optional or Conditional Properties

Property	Description	Sample Value
cmp.issuerDN	Issuer Distinguished Name, if required by Operator CA policy.	/C=SE/O=EJBCA Sample/CN=ManagementCA
cmp.totalTimeout	Maximum total time, in seconds, allowed for a complete CMP transaction. The value must be a positive integer from 1 through 480. The default is 30 seconds.	30
cmp.messageTimeout	Maximum time, in seconds, allowed for one CMP message exchange. The value must be a positive integer from 1 through the configured `cmp.totalTimeout` value and cannot exceed it. The default is 15 seconds.	15
cmp.enableTLS	Set to `true` when the CMP endpoint uses TLS and OCLM must validate the TLS server certificate. Set to `false` when TLS validation is not required.	false
cmp.tlsTrustedCerts	Comma separated list of local TLS trust certificate files. Required when `cmp.enableTLS=true`. Keep this property commented when `cmp.enableTLS=false`.	/opt/oclm-service/operator-ca-tls-root.pem

`configureIssuer.py` validates both timeout values before writing the file. Pressing **Enter** accepts the displayed existing or default value; the script does not write an empty timeout value.

Sample Script Execution Log

The following log shows a sample execution. The CMP endpoint IP address is masked for documentation. Replace all placeholder IP values, Distinguished Names, and file names with customer-specific values during deployment.

```
python3 configureIssuer.py
Operator CA CMP issuerConfig.properties updater
```

```
Target file: /opt/oclm-service/issuerConfig.properties
cmp.server
Definition : Operator CA CMP endpoint used by OCLM to send CMP requests.
Sample : https://operator-ca.example.com:8443/ejbca/publicweb/cmp/oclm
Enter cmp.server [https://operator-ca.example.com:8443/ejbca/publicweb/cmp/
oclm]: http://10.xxx.xxx.xxx:8080/ejbca/publicweb/cmp/occmalias
cmp.recipientDN
Definition : Distinguished Name of the CMP recipient expected by Operator CA.
Sample : /C=SE/O=EJBCA Sample/CN=ManagementCA
Enter cmp.recipientDN [/C=SE/O=EJBCA Sample/CN=ManagementCA]: /C=SE/O=EJBCA
Sample/CN=ManagementCA
cmp.issuerDN
Definition : Distinguished Name of the issuer expected by Operator CA.
Sample : /C=SE/O=EJBCA Sample/CN=ManagementCA
Enter cmp.issuerDN [/C=SE/O=EJBCA Sample/CN=ManagementCA]: /C=SE/O=EJBCA
Sample/CN=ManagementCA
cmp.totalTimeout
Definition : Maximum total time, in seconds, allowed for a complete CMP
transaction (range 1-480).
Sample : 30
Note : Default value is 30 seconds.
Enter cmp.totalTimeout [30]: 30
cmp.messageTimeout
Definition : Maximum time, in seconds, allowed for one CMP message exchange
(range 1-cmp.totalTimeout).
Sample : 15
Enter cmp.messageTimeout [15]: 15
cmp.rootCACerts
Definition : Comma-separated CA certificate paths used to verify certificates
issued by Operator CA.
Sample : /opt/oclm-service/operator-ca-root.pem,/opt/oclm-service/operator-ca-
intermediate.pem
Enter cmp.rootCACerts [/opt/oclm-service/operator-ca-root.pem]: /opt/oclm-
service/ManagementCA.pem
cmp.enableTLS
Definition : Enables TLS validation for the Operator CA HTTPS endpoint.
Sample : true
Enter cmp.enableTLS (true/false) [true]: false
cmp.enableTLS is false, so cmp.tlsTrustedCerts will be written as a commented
property.
Configuration summary for /opt/oclm-service/issuerConfig.properties:
cmp.server=http://10.xxx.xxx.xxx:8080/ejbca/publicweb/cmp/occmalias
cmp.recipientDN=/C=SE/O=EJBCA Sample/CN=ManagementCA
cmp.issuerDN=/C=SE/O=EJBCA Sample/CN=ManagementCA
cmp.totalTimeout=30
cmp.messageTimeout=15
cmp.rootCACerts=/opt/oclm-service/ManagementCA.pem
cmp.enableTLS=false
# cmp.tlsTrustedCerts=/opt/oclm-service/operator-ca-tls-root.pem
Write these changes (y/N): y
Updated: /opt/oclm-service/issuerConfig.properties
```

When `cmp.enableTLS=false`, `cmp.tlsTrustedCerts` may remain commented out. When `cmp.enableTLS=true`, provide a valid `cmp.tlsTrustedCerts path`.

4.6 Configure the Initial OCLM CMP Client Certificate

OCLM uses an initial CMP client certificate and matching private key to establish the first trusted CMP identity between the OCLM node and Operator CA. Operator CA must recognize this certificate as an authorized CMP client before OCLM can request or renew certificates.

`initialIssuedCert.pem` is the initial OCLM CMP client certificate. It identifies the OCLM node to Operator CA and is used for signature-protected CMP requests.

`initialIssuedKey.pem` is the private key that matches `initialIssuedCert.pem`. OCLM uses this private key to sign CMP requests. Keep this file private and readable only by the OCLM service account and authorized administrators.

Together, these files are used for initial, signature-based trust establishment between OCLM and Operator CA. They are different from the Operator CA trust certificates under `/opt/oclm-service`; the Operator CA trust certificates are used to verify issued certificates and, when enabled, the CMP TLS endpoint.

Default file names:

```
/opt/oclm-service/certs/initialIssuedCert.pem
/opt/oclm-service/certs/initialIssuedKey.pem
```

Create the staging directory:

```
STAGE_DIR=/home/admusr/oclm-staging/cmp-identity
mkdir -p "$STAGE_DIR"
chmod 700 "$STAGE_DIR"
```

Copy `initialIssuedCert.pem` and `initialIssuedKey.pem` into $STAGE_DIR using the customer-approved secure transfer method. Then install the files from the staging directory:

```
sudo install -m 0644 "$STAGE_DIR/initialIssuedCert.pem" /opt/oclm-service/
certs/initialIssuedCert.pem
sudo install -m 0600 "$STAGE_DIR/initialIssuedKey.pem" /opt/oclm-service/
certs/initialIssuedKey.pem
sudo chown admusr:admusr /opt/oclm-service/certs/initialIssuedCert.pem /opt/
oclm-service/certs/initialIssuedKey.pem
```

Remove the staged copies after the files are installed:

```
rm -f "$STAGE_DIR/initialIssuedCert.pem" "$STAGE_DIR/initialIssuedKey.pem"
rmdir "$STAGE_DIR" 2>/dev/null || true
```

Validate the certificate and key:

```
openssl x509 -in /opt/oclm-service/certs/initialIssuedCert.pem -noout -
```

```
subject -issuer -dates
openssl pkey -in /opt/oclm-service/certs/initialIssuedKey.pem -noout -check
```

> **Note**
>
> Creation and renewal of the initial trust certificates must be managed manually by the operator. The OCLM node does not create or renew these certificates automatically.

4.7 Start and Enable OCLM

Start OCLM only after Operator CA configuration, CMP identity files, and trust files are in place.

```
sudo systemctl daemon-reload
sudo systemctl start oclm-service
sudo systemctl status --no-pager oclm-service
```

If the service starts successfully and configuration is final for the site, enable the service for future boots.

```
sudo systemctl enable oclm-service
```

Validate that OCLM is listening on TCP 8989:

```
sudo ss -ltnp | grep ':8989'
```

5

Certificate API

The OCLM REST endpoint runs on port 8989 by default.

Base URL:

```
https://<oclm-vm-ip>:8989
```

Endpoint Summary

```
POST /oclm-config/v1/certs
GET /oclm-config/v1/certs/{name-or-uuid}
POST /oclm-config/v1/certs/{name}/renew
```

5.1 Common Request Model

Certificate create and renew requests use the same certificate payload shape for the requested certificate details. Renew also requires `oldCertificate`, which identifies the existing certificate being renewed.

Table 5-1 Certificate Request Payload Fields

Field	Create	Renew	Description
name	Required	Required	Certificate name used by OCLM and by the retrieve API.
uuid	Optional	Not applicable (the renew endpoint uses {name}, not uuid).	Optional client-supplied identifier for create requests. When supplied, use this value with the retrieve API.
renewBefore	Optional	Optional	Renewal threshold value such as 15d. If omitted, the default value is used.
oldCertificate	Not used	Required	Full PEM certificate body of the existing certificate being renewed. Do not provide a filesystem path.
privateKey	Required	Required	New private key generation settings. OCLM generates the private key; clients do not submit the private key in the request.
csr	Required	Required	Requested certificate subject, subject alternative names, validity, key usage, extended key usage, and basic constraints.

Table 5-2 Supported values

Field	Supported Values
privateKey.keyAlgo	RSA
privateKey.keySize	KEYSIZE_2048, KEYSIZE_4096
privateKey.keyEncoding	PEM only. DER requests are rejected with INVALID_INPUT.
csr.keyUsage.keyUsageValues	DIGITAL_SIGNATURE, KEY_ENCIPHERMENT, NON_REPUDIATION
csr.extendedKeyUsage.extendedKeyUsageValues	CLIENT_AUTH, SERVER_AUTH
csr.basicConstraints.basicConstraintsValue	END_ENTITY

5.2 Create Certificate

Perform the following command to create a new certificate:

```
POST /oclm-config/v1/certs
```

Creates a new certificate through CMP initial registration. The operation is asynchronous and returns HTTP 202 Accepted after the request is scheduled.

Sample request payload:

```
{
  "name": "server",
  "renewBefore": "15d",
  "privateKey": {
    "keyAlgo": "RSA",
    "keySize": "KEYSIZE_2048",
    "keyEncoding": "PEM"
  },
  "csr": {
    "subject": {
      "country": "US",
      "state": "CA",
      "location": "Redwood Shores",
      "organization": "Oracle",
      "organizationUnit": "OCLM",
      "commonName": "server.example.com"
    },
    "subjectAltName": {
      "critical": false,
      "dns": ["server.example.com"],
      "ipAddress": ["10.0.0.10"]
    },
    "keyUsage": {
      "critical": false,
      "keyUsageValues": ["DIGITAL_SIGNATURE"]
    },
    "extendedKeyUsage": {
      "critical": false,
```

```
      "extendedKeyUsageValues": ["CLIENT_AUTH", "SERVER_AUTH"]
    },
    "basicConstraints": {
      "critical": false,
      "basicConstraintsValue": "END_ENTITY"
    },
    "days": "390"
  }
}
```

Successful response:

```
202 Accepted
```

```
{
  "name": "server.example.com",
  "status": "ACCEPTED"
}
```

When `uuid` is present in the response, use it with the retrieve API. If not, use the `name`.

5.3 Retrieve Certificate

Perform the following command to retrieve certificate:

```
GET /oclm-config/v1/certs/{name-or-uuid}
```

Retrieves the generated certificate and private key for a previous create request. This API does not use a request payload. Clients can retrieve by name when the create request omitted `uuid`.

Certificates can be retrieved only once because OCLM does not store end entity certificates on the OCLM node.

Successful response:

```
200 OK
```

```
{
  "cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n",
  "key": "-----BEGIN PRIVATE KEY-----\n...\n-----END PRIVATE KEY-----\n"
}
```

If the certificate is not ready or no generated files exist, the response is 404 IO_READ_ERROR.

5.4 Renew Certificate

Perform the following command to renew certificate:

```
POST /oclm-config/v1/certs/{name}/renew
```

Renews an existing certificate through CMP Key Update Request. This operation is synchronous and returns the renewed certificate and new private key in the response.

For renew, path `{name}` must match the request body `name`. The `oldCertificate` value must contain the PEM certificate body, not a filesystem path.

Sample request payload:

```
{
  "name": "server",
  "oldCertificate": "-----BEGIN CERTIFICATE-----\n...\n-----END
CERTIFICATE-----\n",
  "renewBefore": "15d",
  "privateKey": {
    "keyAlgo": "RSA",
    "keySize": "KEYSIZE_2048",
    "keyEncoding": "PEM"
  },
  "csr": {
    "subject": {
      "country": "US",
      "state": "CA",
      "location": "Redwood Shores",
      "organization": "Oracle",
      "organizationUnit": "OCLM",
      "commonName": "server.example.com"
    },
    "subjectAltName": {
      "critical": false,
      "dns": ["server.example.com"],
      "ipAddress": ["10.0.0.10"]
    },
    "keyUsage": {
      "critical": false,
      "keyUsageValues": ["DIGITAL_SIGNATURE"]
    },
    "extendedKeyUsage": {
      "critical": false,
      "extendedKeyUsageValues": ["CLIENT_AUTH", "SERVER_AUTH"]
    },
    "basicConstraints": {
      "critical": false,
      "basicConstraintsValue": "END_ENTITY"
    },
    "days": "390"
  }
}
```

Successful response:

```
200 OK
```

```
{
  "cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n",
  "key": "-----BEGIN PRIVATE KEY-----\n...\n-----END PRIVATE KEY-----\n"
}
```

5.5 API Failure Handling

The following table provides the API Failure response codes and error messages:

Table 5-3 API Failure Response Codes and Error Messages

Status	Error Code	Typical Cause
400	INVALID_INPUT	Request validation failed, a required field is missing or invalid, an unsupported enum value was supplied, DER key encoding was requested, or the renew request name does not match the path name.
400	MISSING_CERT_TO_BE_RENEWED	Renew request does not include a non-blank `oldCertificate` value.
400	MAX_CERT_LIMIT_REACHED	The maximum number of accepted certificate creation requests has been reached.
404/500	IO_READ_ERROR	Retrieve could not find the generated certificate/key after waiting, or the generated file exists but could not be read.
409	ERR_REQUEST_REJECTED	Certificate creation job was rejected during scheduling.
500	INVALID_CONFIG	Required runtime or issuer configuration is missing or invalid.
500	CMP_COMMAND_FAILED	CMP request failed or Operator CA rejected the request.
500	CMP_COMMAND_TIMEOUT	CMP request timed out.
500	PROCESS_START_FAILURE	CMP command process could not be started.

Problem responses use a common JSON schema:

```
{
  "title": "Invalid input",
  "status": 400,
  "detail": "Invalid input",
  "errorCode": "INVALID_INPUT",
  "invalidParams": []
}
```

6
Operations

6.1 Service Commands

Perform the following `systemd` commands to manage the OCLM service:

```
sudo systemctl start oclm-service
sudo systemctl stop oclm-service
sudo systemctl restart oclm-service
sudo systemctl status --no-pager oclm-service
```

Logs

Use the following commands to collect /var/log/oclm.log and systemd journal output for troubleshooting:

```
sudo tail -n 200 /var/log/oclm.log
sudo journalctl -u oclm-service --no-pager -n 100
```

6.2 Configuration Change Procedure

Oracle recommends using the provided configuration script for Operator CA configuration changes:

- Stop new API activity during the change window.
- Back up the file being changed.
- Run the configuration script or update the required file.
- Validate referenced file paths and permissions.
- Restart OCLM if the changed setting affects service startup, HTTPS serving, or CMP execution.
- Confirm the service is healthy after the change.

Example for `issuerConfig.properties` changes:

```
sudo cp -p /opt/oclm-service/issuerConfig.properties /opt/oclm-service/
issuerConfig.properties.bak.$(date -u +%Y%m%d%H%M%S)
cd /opt/oclm-service
python3 configureIssuer.py
sudo chown admusr:admusr /opt/oclm-service/issuerConfig.properties
sudo chmod 640 /opt/oclm-service/issuerConfig.properties
```

```
sudo systemctl restart oclm-service
sudo systemctl status --no-pager oclm-service
```

6.3 HTTPS TLS Certificate Change Procedure

OCLM HTTPS serving files are:

```
/opt/oclm-service/bin/server.crt.pem
```

```
/opt/oclm-service/bin/server.key.pem
```

`server.crt.pem` is the HTTPS certificate for the OCLM REST endpoint. `server.key.pem` is the matching HTTPS private key. The HTTPS private key should be provided in a service-startup compatible PEM format. Protect `server.key.pem` with 0600 permissions and `admusr:admusr` ownership.

Restart OCLM after replacing the HTTPS serving files.

Run the following command to create the staging directory:

```
STAGE_DIR=/home/admusr/oclm-staging/https-tls
mkdir -p "$STAGE_DIR"
chmod 700 "$STAGE_DIR"
```

Copy the new `server.crt.pem` and `server.key.pem` into $STAGE_DIR using the customer-approved secure transfer method.

Validate the staged files:

```
openssl x509 -in "$STAGE_DIR/server.crt.pem" -noout -subject -issuer -dates
openssl pkey -in "$STAGE_DIR/server.key.pem" -noout -check
```

Stop OCLM and back up the existing HTTPS files by renaming them:

```
sudo systemctl stop oclm-service
sudo systemctl status --no-pager oclm-service
sudo mv /opt/oclm-service/bin/server.crt.pem /opt/oclm-service/bin/
server.crtbk.pem
sudo mv /opt/oclm-service/bin/server.key.pem /opt/oclm-service/bin/
server.keybk.pem
```

Install the new HTTPS TLS files:

```
sudo install -m 0640 "$STAGE_DIR/server.crt.pem" /opt/oclm-service/bin/
server.crt.pem
sudo install -m 0600 "$STAGE_DIR/server.key.pem" /opt/oclm-service/bin/
server.key.pem
```

```
sudo chown admusr:admusr /opt/oclm-service/bin/server.crt.pem /opt/oclm-
service/bin/server.key.pem
```

Remove the staged copies:

```
rm -f "$STAGE_DIR/server.crt.pem" "$STAGE_DIR/server.key.pem"
rmdir "$STAGE_DIR" 2>/dev/null || true
```

Start and validate OCLM:

```
sudo systemctl start oclm-service
sudo systemctl status --no-pager oclm-service
sudo ss -ltnp | grep ':8989'
sudo tail -n 200 /var/log/oclm.log
```

Restore the previous self-signed HTTPS files if required:

```
sudo systemctl stop oclm-service
sudo mv /opt/oclm-service/bin/server.crtbk.pem /opt/oclm-service/bin/
server.crt.pem
sudo mv /opt/oclm-service/bin/server.keybk.pem /opt/oclm-service/bin/
server.key.pem
sudo chown admusr:admusr /opt/oclm-service/bin/server.crt.pem /opt/oclm-
service/bin/server.key.pem
sudo chmod 0640 /opt/oclm-service/bin/server.crt.pem
sudo chmod 0600 /opt/oclm-service/bin/server.key.pem
sudo systemctl start oclm-service
sudo systemctl status --no-pager oclm-service
```

6.4 Configured OCLM User Password Change Procedure

`admusr` is the OCLM administrative/service OS user. Password change is used to maintain restricted administrative access to the OCLM VM. Do not pass the password on the command line. OCLM service restart is not required for password-only change. Change the password interactively:

```
sudo passwd admusr
```

After changing the password, open a new login session and verify access with the updated credential before closing the existing administrative session.

7

Upgrade and Patch Mechanism

OCLM supports two customer maintenance patterns:

Table 7-1 OCLM Upgrade and Patch Update Methods

Maintenance Type	When to Use	Main Action
Patch update	Use when Oracle supplies an OCLM application wheel and required patch dependency files for the same VM release.	Stop OCLM, back up the previous wheel packages, copy the Oracle-provided patch files into the required OCLM folders, run `/opt/oclm-service/bin/upgrade.sh`, and validate the service.
Full upgrade	Use when Oracle supplies a new OCLM VM/VMDK image, base image update, or release that requires redeployment.	Back up customer-specific files from the old VM, deploy a new VM, restore only customer-specific configuration and certificate material, then validate service behavior.

Do not copy old release binaries, old service scripts, the old runtime directory, old patch dependency files, or old `systemd` unit files into a new VM. A full upgrade should restore only customer-specific configuration, trust files, and identity files. The new VM image supplies the release-specific application files.

7.1 Patch Update on the Existing VM

A patch update is an in-place application package update. It replaces the OCLM application wheel and dependency wheel files used by the existing virtual environment. It does not update customer configuration, Operator CA trust files, CMP identity files, or HTTPS serving certificate files.

Oracle provides the patch as a .tar archive with the following contents:

```
oclm-patch-VERSION.tar
├── oclm_service-VERSION-py3-none-any.whl
└── dependencies/
    └── *.whl
```

Table 7-2 Patch Update Package Artifacts

Artifact	Required	Purpose
oclm-patch-VERSION.tar	Mandatory	Oracle-provided patch archive containing the application wheel and dependency wheels.
oclm_service-VERSION-py3-none-any.whl	Mandatory	OCLM application package.

Table 7-2 (Cont.) Patch Update Package Artifacts

Artifact	Required	Purpose
dependencies/*.whl	Mandatory	Offline dependency wheel files required by the OCLM application package.
/opt/oclm-service/bin/upgrade.sh	Mandatory	Patch helper already present on the OCLM VM.

> **Note**
>
> The OCLM VM does not include the tar utility. Extract the archive on an approved administration host and securely copy the extracted files to the OCLM VM.

Extract the patch on the administration host

Verify the Oracle-provided archive according to the release-specific checksum instructions. Extract and inspect the patch package on an approved administration host:

```
mkdir -p oclm-patch-VERSION
tar -xf oclm-patch-VERSION.tar -C oclm-patch-VERSION
```

Confirm that the extracted package contains exactly one OCLM application wheel and all required dependency wheels.

Copy the extracted patch to the OCLM VM

Use a customer approved secure transfer method to copy the extracted directory to the OCLM VM. The following example uses scp:

```
scp -r oclm-patch-VERSION   admusr@<oclm-node-ip>:/home/admusr/
```

Validate the copied patch files

Log in to the OCLM VM and confirm that the application and dependency wheel files are present:

```
ssh admusr@<oclm-node-ip>

PATCH_DIR=/home/admusr/oclm-patch-VERSION

find "$PATCH_DIR" -maxdepth 1 -name 'oclm_service-*.whl' -print
find "$PATCH_DIR/dependencies" -name '*.whl' -print
```

Stop OCLM and back up the existing wheel files

Stop the OCLM service and create a backup of the currently installed application and dependency wheel files. Retain this backup until patch acceptance is complete.

```
sudo systemctl stop oclm-service
sudo systemctl status --no-pager oclm-service

BACKUP_DIR=/home/admusr/oclm-patch-backup-$(date -u +%Y%m%d%H%M%S)

sudo mkdir -p "$BACKUP_DIR/wheels"
sudo cp -p /opt/oclm-service/oclm_service-*.whl "$BACKUP_DIR/"
sudo cp -p /opt/oclm-service/wheels/*.whl "$BACKUP_DIR/wheels/"
sudo chown -R admusr:admusr "$BACKUP_DIR"

ls -l "$BACKUP_DIR"
ls -l "$BACKUP_DIR/wheels"
```

Do not continue if the backup fails. This backup contains only application and dependency wheel files; it does not include customer configuration, Operator CA trust files, CMP identity files, or HTTPS serving certificate files.

Copy the New Patch Artifacts

Remove the previous wheel files and copy the extracted patch artifacts to the locations used by `upgrade.sh`:

```
PATCH_DIR=/home/admusr/oclm-patch-VERSION

sudo install -d -m 0755 /opt/oclm-service/wheels

sudo rm -f /opt/oclm-service/oclm_service-*.whl
sudo rm -f /opt/oclm-service/wheels/*.whl

sudo install -m 0644   "$PATCH_DIR"/oclm_service-*.whl   /opt/oclm-service/

sudo cp -p   "$PATCH_DIR"/dependencies/*.whl   /opt/oclm-service/wheels/

sudo chown admusr:admusr   /opt/oclm-service/oclm_service-*.whl   /opt/oclm-
service/wheels/*.whl
```

Run the Patch

Run the patch helper from the OCLM VM:

```
sudo bash /opt/oclm-service/bin/upgrade.sh
```

The helper installs the OCLM application and its dependencies from the local wheel files. After successful installation, it restarts `oclm-service.service`.

Expected successful output includes:

```
Upgraded OCLM Python wheel in /opt/oclm-service/venv using /opt/oclm-service/
```

```
oclm_service-VERSION-py3-none-any.whl
Restarted oclm-service.service
```

Validate the Patch

Validate the service status, listener, installed package version, and recent logs:

```
sudo systemctl status --no-pager oclm-service
sudo ss -ltnp | grep ':8989'
sudo tail -n 200 /var/log/oclm.log
```

Perform an approved certificate create and retrieve test before accepting the patch. After acceptance, remove the staged patch files according to the customer retention policy. Retain the patch backup until rollback is no longer required.

7.2 Patch Rollback

Rollback restores the previous OCLM application wheel and dependency wheel files, then reruns `upgrade.sh` so the existing virtual environment is reinstalled from those previous artifacts.

```
BACKUP_DIR=/home/admusr/oclm-patch-backup-TIMESTAMP

sudo systemctl stop oclm-service

sudo rm -f /opt/oclm-service/oclm_service-*.whl
sudo rm -f /opt/oclm-service/wheels/*.whl

sudo cp -p "$BACKUP_DIR"/oclm_service-*.whl /opt/oclm-service/
sudo cp -p "$BACKUP_DIR"/wheels/*.whl /opt/oclm-service/wheels/

sudo chown admusr:admusr   /opt/oclm-service/oclm_service-*.whl   /opt/oclm-
service/wheels/*.whl

sudo bash /opt/oclm-service/bin/upgrade.sh
sudo systemctl status --no-pager oclm-service
sudo ss -ltnp | grep ':8989'
sudo tail -n 200 /var/log/oclm.log
```

If rollback does not restore expected service behavior, restore the VM snapshot taken before the patch or contact support.

7.3 Full Upgrade Using a New VM

A full upgrade uses a new OCLM VMDK or QCOW2 image. Use this procedure when the release notes require a new image, when the base OS image changes, when the VM must be rebuilt, or when support directs a redeployment.

Perform the following recommended full upgrade procedure:

1. Back up customer-specific files from the old OCLM VM.

2. Deploy a new OCLM VM from the approved new image.
3. Restore only customer-specific configuration and certificate material.
4. Validate configuration, service startup, Operator CA connectivity, and API behavior.
5. Activate approved API clients on the new VM.
6. Retain the old VM powered off until customer acceptance is complete.

The following files must be backed up from the existing VM before a full upgrade:

Table 7-3 Full Upgrade Backup Set

Path	Required	Restore to New VM	Notes
/opt/oclm-service/issuerConfig.properties	Mandatory	Yes	Operator CA CMP configuration. Run `configureIssuer.py` after restore if endpoint, DN, TLS, or trust paths change.
/opt/oclm-service/operator-ca-root.pem	Mandatory	Yes	Operator CA trust anchor used by `ecmp.rootCACerts`.
/opt/oclm-service/operator-ca-intermediate.pem	Conditional	Yes	Required only when the Operator CA chain uses an intermediate CA.

Table 7-3 (Cont.) Full Upgrade Backup Set

Path	Required	Restore to New VM	Notes
/opt/oclm-service/operator-ca-tls-root.pem	Conditional	Yes, if configured	Required only when `ecmp.enableTLS=true`.
/opt/oclm-service/certs/initialIssuedCert.pem	Mandatory	Yes	Initial OCLM CMP client certificate trusted by Operator CA.
/opt/oclm-service/certs/initialIssuedKey.pem	Mandatory	Yes	Matching private key for the initial CMP client certificate. Protect this file as secret data.
/opt/oclm-service/bin/server.crt.pem	Conditional	Yes, if customer-specific	HTTPS certificate for the OCLM REST endpoint, when supplied by the site.

Table 7-3 (Cont.) Full Upgrade Backup Set

Path	Required	Restore to New VM	Notes
/opt/oclm-service/bin/server.key.pem	Conditional	Yes, if customer-specific	HTTPS private key for the OCLM REST eendpoint. Protect this file as secret data.
/var/log/oclm.log	Optional	No	Retain only for audit or troubleshooting. Do not restore old logs over the new service log.

Do not restore the old VM application wheel, patch dependency files or OS packages into the new VM. Before taking the backup, stop the OCLM service.

```
sudo systemctl stop oclm-service
sudo systemctl status --no-pager oclm-service
```

Generated certificate files under `/opt/oclm-service/certs` are transient. If unretrieved generated material exists during activation, retrieve it through the API before decommissioning the old VM.

Create the full upgrade backup

Perform the following steps on the old OCLM VM after stopping the OCLM service:

1. Create the backup directory

Create a timestamped backup directory under `/home/admusr`. The subdirectories keep configuration, CMP identity, HTTPS serving files, and logs separated so they are easy to review before restore.

```
BACKUP_ROOT=/home/admusr/oclm-full-upgrade-backup-$(hostname)-$(date -u +
%Y%m%d%H%M%S)
sudo mkdir -p $BACKUP_ROOT/service $BACKUP_ROOT/
certs $BACKUP_ROOT/bin $BACKUP_ROOT/logs
```

2. Back up Operator CA configuration and trust files
   Back up `issuerConfig.properties` and the Operator CA trust files. These files are required on the new VM so OCLM can continue to reach Operator CA and validate certificates issued by Operator CA. The intermediate and TLS trust files are copied only when they exist on the old VM.

```
sudo cp -p /opt/oclm-service/issuerConfig.properties $BACKUP_ROOT/service/
sudo cp -p /opt/oclm-service/operator-ca-root.pem $BACKUP_ROOT/service/
[ -f /opt/oclm-service/operator-ca-intermediate.pem ] && sudo cp -p /opt/
oclm-service/operator-ca-intermediate.pem $BACKUP_ROOT/service/
[ -f /opt/oclm-service/operator-ca-tls-root.pem ] && sudo cp -p /opt/oclm-
service/operator-ca-tls-root.pem $BACKUP_ROOT/service/
```

3. Back up the initial CMP client identity
   Back up the initial CMP client certificate and private key. These files are required because the new OCLM VM uses the same trusted CMP client identity when sending signature-protected CMP requests to Operator CA. Protect the private key file and any archive that contains it.

```
sudo cp -p /opt/oclm-service/certs/initialIssuedCert.pem $BACKUP_ROOT/
certs/
sudo cp -p /opt/oclm-service/certs/initialIssuedKey.pem $BACKUP_ROOT/certs/
```

4. Back up optional HTTPS serving files and logs
   Back up the HTTPS serving certificate and private key only when they exist are customer-specific. Back up the current log only for audit or troubleshooting reference. Do not restore logs over the new VM log.

```
[ -f /opt/oclm-service/bin/server.crt.pem ] && sudo cp -p /opt/oclm-
service/bin/server.crt.pem $BACKUP_ROOT/bin/
[ -f /opt/oclm-service/bin/server.key.pem ] && sudo cp -p /opt/oclm-
service/bin/server.key.pem $BACKUP_ROOT/bin/
[ -f /var/log/oclm.log ] && sudo cp -p /var/log/oclm.log $BACKUP_ROOT/logs/
```

5. Create and verify the backup archive
   Create a compressed archive under `/home/admusr` and verify that the archive was created. Copy this archive to customer-approved secure storage before decommissioning or powering off the old VM.

```
sudo tar -C /home/admusr -czf $BACKUP_ROOT.tar.gz $(basename $BACKUP_ROOT)
sudo ls -lh $BACKUP_ROOT.tar.gz
```

The backup archive contains private key material. Restrict access to authorized deployment administrators only.

Deploy the new VM

Deploy the new OCLM VM from the approved image:

- Verify the new VM or VMDK artifact checksum.
- Create the VM using the approved CPU, memory, disk, and network settings.

Restore customer-specific files

Copy the backup archive to the new VM before running these commands. Extract it under `/home/admusr`, then restore only the customer-specific files from the backup.

1. Stage and extract the backup archive
   Set the backup archive path and extract it into a restore working directory. The `--strip-components=1` option removes the top-level backup directory from the archive so the `service`, `certs`, `bin`, and `logs` directories are directly under `$RESTORE_ROOT`.

   ```
   BACKUP_ARCHIVE=/home/admusr/oclm-full-upgrade-backup-<old-host>-
   <timestamp>.tar.gz
   RESTORE_ROOT=/home/admusr/oclm-restore
   sudo mkdir -p $RESTORE_ROOT
   sudo tar -xzf $BACKUP_ARCHIVE -C $RESTORE_ROOT --strip-components=1
   ```

2. Restore Operator CA configuration and trust files
   Restore `issuerConfig.properties` and Operator CA trust files to `/opt/oclm-service`. The root CA file is mandatory. Intermediate CA and TLS trust files are restored only when they were present in the backup.

   ```
   sudo install -m 0640 $RESTORE_ROOT/service/issuerConfig.properties /opt/
   oclm-service/issuerConfig.properties
   sudo install -m 0644 $RESTORE_ROOT/service/operator-ca-root.pem /opt/oclm-
   service/operator-ca-root.pem
   [ -f $RESTORE_ROOT/service/operator-ca-intermediate.pem ] && sudo install -
   m 0644 $RESTORE_ROOT/service/operator-ca-intermediate.pem /opt/oclm-
   service/operator-ca-intermediate.pem
   [ -f $RESTORE_ROOT/service/operator-ca-tls-root.pem ] && sudo install -m
   0644 $RESTORE_ROOT/service/operator-ca-tls-root.pem /opt/oclm-service/
   operator-ca-tls-root.pem
   ```

3. Restore the initial CMP client identity
   Restore the initial CMP client certificate and private key. These files allow the new OCLM VM to authenticate to Operator CA using the trusted CMP client identity from the old VM. The private key is installed with 0600 permissions.

   ```
   sudo install -d -m 0700 -o admusr -g admusr /opt/oclm-service/certs
   sudo install -m 0644 $RESTORE_ROOT/certs/initialIssuedCert.pem /opt/oclm-
   service/certs/initialIssuedCert.pem
   sudo install -m 0600 $RESTORE_ROOT/certs/initialIssuedKey.pem /opt/oclm-
   service/certs/initialIssuedKey.pem
   ```

4. Restore optional HTTPS serving files

Restore the HTTPS serving certificate and private key only if they exist in the backup. These files are needed only when the site uses customer-specific HTTPS serving material for the OCLM REST endpoint.

```
[ -f $RESTORE_ROOT/bin/server.crt.pem ] && sudo install -m
0640 $RESTORE_ROOT/bin/server.crt.pem /opt/oclm-service/bin/server.crt.pem
[ -f $RESTORE_ROOT/bin/server.key.pem ] && sudo install -m
0600 $RESTORE_ROOT/bin/server.key.pem /opt/oclm-service/bin/server.key.pem
```

5. Set ownership on restored files
   Set ownership to `admusr:admusr` for the restored configuration, trust, identity, and optional HTTPS serving files so the OCLM service can read them with the intended permissions.

```
sudo chown admusr:admusr /opt/oclm-service/issuerConfig.properties
sudo chown admusr:admusr /opt/oclm-service/operator-ca-root.pem
[ -f /opt/oclm-service/operator-ca-intermediate.pem ] && sudo chown
admusr:admusr /opt/oclm-service/operator-ca-intermediate.pem
[ -f /opt/oclm-service/operator-ca-tls-root.pem ] && sudo chown
admusr:admusr /opt/oclm-service/operator-ca-tls-root.pem
sudo chown admusr:admusr /opt/oclm-service/certs/
initialIssuedCert.pem /opt/oclm-service/certs/initialIssuedKey.pem
[ -f /opt/oclm-service/bin/server.crt.pem ] && sudo chown
admusr:admusr /opt/oclm-service/bin/server.crt.pem
[ -f /opt/oclm-service/bin/server.key.pem ] && sudo chown
admusr:admusr /opt/oclm-service/bin/server.key.pem
```

If the new VM uses a different Operator CA endpoint, IP address, TLS setting, or certificate path, run the configuration script after restore:

```
cd /opt/oclm-service
python3 configureIssuer.py
sudo chown admusr:admusr /opt/oclm-service/issuerConfig.properties
sudo chmod 640 /opt/oclm-service/issuerConfig.properties
```

Validate the new VM before acceptance

Validate configuration and files before exposing the new VM to production API clients.

```
sudo test -f /opt/oclm-service/issuerConfig.properties && echo "exists" ||
echo "missing"
sudo test -f /opt/oclm-service/operator-ca-root.pem && echo "exists" || echo
"missing"
sudo test -f /opt/oclm-service/certs/initialIssuedCert.pem && echo "exists"
|| echo "missing"
sudo test -f /opt/oclm-service/certs/initialIssuedKey.pem && echo "exists" ||
echo "missing"
sudo systemctl daemon-reload
sudo systemctl start oclm-service
sudo systemctl status --no-pager oclm-service
sudo ss -ltnp | grep ':8989'
sudo tail -n 200 /var/log/oclm.log
```

Service activation and acceptance

After validation succeeds, activate the new OCLM VM as follows:

1. Enable the service for future boots
   Enable `oclm-service` only when the new VM is accepted as the active OCLM node.

```
sudo systemctl enable oclm-service
sudo systemctl status --no-pager oclm-service
```

2. Monitor OCLM during the acceptance window
   Monitor the service and recent logs after API clients or DSD/SDS integration are moved to the new VM. No repeated startup, TLS, CMP, or certificate validation errors should be present.

```
sudo systemctl status --no-pager oclm-service
sudo tail -n 200 /var/log/oclm.log
sudo journalctl -u oclm-service --no-pager -n 100
```

3. Remove the old backups after approval
   Remove old backup material only after new OCLM VM service remains healthy.

```
sudo rm -f /home/admusr/oclm-full-upgrade-backup-<old-host>-
<timestamp>.tar.gz
sudo rm -rf /home/admusr/oclm-full-upgrade-backup-<old-host>-<timestamp>
```

8
Troubleshooting

Use `/var/log/oclm.log` and `systemd` status output as the first troubleshooting sources.

```
sudo systemctl status --no-pager oclm-service
sudo tail -n 200 /var/log/oclm.log
```

The following table provides common checks:

Table 8-1 Common checks

Symptom	Check
Service does not start	Confirm `issuerConfig.properties`, initial CMP identity files, and required trust files exist and are readable by `admusr`.
CMP request fails	Confirm `cmp.server`, `cmp.recipientDN`, `cmp.rootCACerts`, TLS settings, network reachability, and Operator CA policy.
API client cannot connect	Confirm TCP 8989 is listening and customer network policy permits access.
Certificate validation fails	Confirm system time, Operator CA trust chain, and the initial CMP identity certificate validity period.

9

Security Considerations

Protect private key files and backup archives that contain private key material. Limit access to authorized administrators.

Use customer-approved secure transfer mechanisms when moving trust files, identity files, TLS serving files, patch artifacts, and backup archives between systems. Remove staged copies after files are installed.

A
Out of Scope

The following capabilities are outside the scope of this release:

- DSR and SDS certificate recovery and revocation are not supported through the OCLM node.
- Initial Operator CA configuration is supported only through the OCLM node CLI, graphical user interface (GUI) is not supported.
- Certificates for the VNFM and ILOM interfaces are not supported. Only web certificates for DSR and SDS at the A and B levels are supported.
- Deployment of the OCLM node through VNFM is not supported. The OCLM nodes must be installed and configured manually.