Oracle® Communications Billing and Revenue Management

Cloud Native Deployment Guide



Release 16.26.0
G54446-02
October 2026

ORACLE

Oracle Communications Billing and Revenue Management Cloud Native Deployment Guide, Release 16.26.0

G54446-02

Copyright © 2019, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

About This Content

Part I Overview of BRM Cloud Native

1 Overview of the BRM Cloud Native Deployment

About the BRM Cloud Native Deployment 1
BRM Cloud Native Deployment Architecture 1

2 About the BRM Cloud Native Deployment Packages

Overview of the BRM Cloud Native Deployment Package 1
About BRM Pods 1
About Client Pods and Images 3
About BRM PVCs and Pods 5

Part II Getting Started with BRM Cloud Native Deployment

3 About Configuring and Deploying Your BRM Cloud Native Environment

About Configuring and Deploying BRM Cloud Native 1
High-Level Installation Tasks 2

4 Setting Up Prerequisite Software

BRM Cloud Native Prerequisite Tasks 1
Software Compatibility 2
Creating a Kubernetes Cluster 2
Installing Podman 3
Installing Helm 4
Creating and Configuring Your BRM Database 4
Installing an External Provisioner 6
Installing WebLogic Kubernetes Operator 6

Copyright © 2019, 2026, Oracle and/or its affiliates.

Exposing BRM Services Outside the Kubernetes Cluster 7
Setting Up ECE Cloud Native Ingress and Egress Flows 8

5 Preparing Your BRM Cloud Native Environment

Tasks for Preparing Your BRM Cloud Native Environment 1
Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files 1
Pulling BRM Images from the Oracle Container Registry 3
Downloading BRM Images from Oracle Software Delivery Website 6
Pulling WebLogic Images for PDC, Billing Care, Billing Care REST API, and Business Operations Center 9

Part III Configuring and Deploying BRM Cloud Native

6 Deploying the BRM Database Schema

Deploying BRM with a New Database Schema 1
Deploying BRM with an Existing Schema 8

7 Configuring BRM Server and PDC Services

About Configuring BRM Cloud Native Services 1
Creating Secrets for Docker Registry Authorization 1
Managing Wallet and KeyStore Certificates 2
- Selecting the TLS Wallet Source 3
- Configuring BRM Client Applications for Two-Way TLS 3

Configuring Global Values 3
Specifying the BRM Services to Deploy 6
Configuring the BRM Server 7
Configuring BRM for a Multischema Database 46
Configuring Pricing Design Center 49
- Adding PDC Keys for oc-cn-helm-chart 50
- Adding PDC Keys for oc-cn-op-job-helm-chart 57
- Configuring SSL for the PDC WebLogic Domain 66
- Setting Up SSO for PDC Cloud Native 67

Configuring Pipeline Configuration Center 69
- Adding Pipeline Configuration Center Keys for oc-cn-op-job-helm-chart 69
- Adding Pipeline Configuration Center Keys for oc-cn-helm-chart 73
- About PCC Volume Mounts 76
- Creating a WebLogic Domain and Installing the PCC Application 76
- Setting Up SSO for Pipeline Configuration Center 77
- Setting Up Local Users and Groups for PCC 78

Copyright © 2019, 2026, Oracle and/or its affiliates.

Starting and Stopping WebLogic Servers 79
Configuring SSL in PCC 80

8 Configuring REST Services

Configuring BRM REST Services Manager 1
Generating an SSL Certificate for BRM REST Services Manager 1
Configuring the SDK (Optional) 2
Connecting to a Separate BRM Cluster 2
Adding BRM REST Services Manager Keys 2
Configuring Policies for API Authorization 9
Configuring and Adding Custom Mapper Files 10
Publishing Notifications for External Applications 12
Notification Framework Prerequisites 13
Setting Up Connection Profiles 14
Configuring Consumers for Kafka Topics 16
Configuring Consumers for Database Queues 19
Setting Up Producers 20
Performing Verification and Sanity Checks 21
Configuring PDC REST Services Manager 21
Adding PDC REST Services Manager Keys 22
Configuring OAuth Authentication in PDC REST Services Manager 26
Configuring Requests to the Enterprise Product Catalog 27
Enabling TLS in PDC REST Services Manager 29
Enabling T3S in PDC REST Services Manager 30
Configuring Mapping of TMF620 priceType to BRM Events 30

9 Configuring the BRM Client Services

About Configuring Your BRM Client Services 1
Configuring Business Operations Center 2
Adding Business Operations Center Keys for oc-cn-op-job-helm-chart 3
Adding Business Operations Center Keys for oc-cn-helm-chart 9
Updating Infranet.properties for Business Operations Center 13
Adding Custom Configuration to Deployment Workflow for Business Operations Center 13
About Business Operations Center Volume Mounts 14
Creating a WebLogic Domain and Installing the Business Operations Center Application 14
Setting Up SSO for Business Operations Center 15
Setting Up Local Users and Groups for Business Operations Center 16
Starting and Stopping WebLogic Servers 17
Adding Authorization for a Custom Job Resource 18
Configuring Collections Configuration Center 18

Copyright © 2019, 2026, Oracle and/or its affiliates.

Adding Collections Configuration Center Keys for oc-cn-helm-chart 18
Configuring Billing Care 22
Adding Billing Care Keys for oc-cn-op-job-helm-chart 23
Adding Billing Care Keys for oc-cn-helm-chart 31
Updating Infranet.properties for Billing Care 38
Adding Custom Configuration to Deployment Workflow for Billing Care 39
About Billing Care Volume Mounts 39
Creating a WebLogic Domain and Installing the Billing Care Application 40
Setting Up SSO for Billing Care 40
Setting Up Local Users and Groups for Billing Care 42
Starting and Stopping WebLogic Servers 42
Configuring the Billing Care REST API 43
Adding Billing Care REST API Keys for oc-cn-op-job-helm-chart 44
Adding Billing Care REST API Keys for oc-cn-helm-chart 46
Updating Infranet Properties for the Billing Care REST API 49
Adding Custom Configuration to Deployment Workflow for Billing Care REST API 49
About Billing Care REST API Volume Mounts 50
Creating a WebLogic Domain and Installing the Billing Care REST API 50
Setting Up Local Users and Groups for Billing Care REST API 51
Starting and Stopping WebLogic Servers 52

10 Configuring ECE Services

Adding Elastic Charging Engine Keys 1
Configuring Gateway Service Types and NodePorts 103
Enabling SSL in Elastic Charging Engine 104
Enabling SSL for the HTTP Gateway Server 105
Connecting ECE Cloud Native to an SSL-Enabled Database 106
Using the ECE Helm Chart to Create Kubernetes Secrets 108
Using Pre-created Kubernetes Secrets 108
About Elastic Charging Engine Volume Mounts 109
Loading Custom Diameter AVP 110
Generating CDRs for Unrated Events 111
Scaling the cdrgateway and cdrFormatter Pods 114
Configuring ECE to Support Prepaid Usage Overage 115
Recording Failed ECE Usage Requests 116
Loading BRM Configuration XML Files 116
Setting Up Notification Handling in ECE 116
Creating an Apache Kafka Notification Topic 117
Creating an Oracle WebLogic Notification Queue 118
Configuring ECE for a Multischema BRM Environment 120
Configuring ECE for Overload Protection 123

Copyright © 2019, 2026, Oracle and/or its affiliates.

Configuring Direct Loading Using the ECE RE Manager Plug-in 123

11 Deploying and Configuring the Mapper Framework

Overview of the Mapper Framework 1
Deploying Custom Mapper Files 1
Configuring the Mapper in BRM Helm Charts 2
Mapper Framework Configuration Keys 3
Example Usage in override-values.yaml 3
Best Practices for Managing Mapper Files 4

12 Deploying BRM Cloud Native Services

Deploying BRM Cloud Native Services 1

13 Deploying into Oracle Cloud Infrastructure

Deploying into Oracle Cloud Infrastructure 1

14 Uninstalling Your BRM Cloud Native Deployment

Uninstalling Your BRM Cloud Native Deployment 1
Uninstalling Selected BRM Cloud Native Services 1

Part IV Customizing BRM Cloud Native

15 Customizing BRM Cloud Native Services

Customizing BRM Server 1
Customizing Billing Care 5
Customizing ECE 6

16 Building Your Own Images

Building BRM Server Images 1
Building Your BRM Server Base Image 2
Building Images of BRM Server Components 3
Building Web Services Manager Images 4
Containerization of Email Data Manager 5
Containerization of Roaming Pipeline 6
Building and Deploying Vertex Manager 6
Deploying with Vertex Communications Tax Q Series 6

Copyright © 2019, 2026, Oracle and/or its affiliates.

Deploying with Vertex Sales Tax Q Series 8
Building BRM REST Services Manager Images 10
Building PDC REST Services Manager Images 12
Building PDC Images 12
Building Pipeline Configuration Center Images 13
Pulling the Fusion Middleware Infrastructure Image 14
Building the Pipeline Configuration Center Image 14
Building Billing Care Images 14
Pulling the Fusion Middleware Infrastructure Image 14
Building the Billing Care Image 14
Building Business Operations Center Images 15
Building Collections Configuration Center Images 15

Part V Upgrading BRM Cloud Native

17 Upgrading Your BRM Cloud Native Environment

Tasks for the BRM Cloud Native Upgrade 1
Upgrading Your Database Schema 2
Upgrading Your BRM Cloud Native Services 3
Upgrading Your ECE Cloud Native Services 5
Upgrading ECE Cloud Native to the Latest Interim Patch 7
Upgrading Your PDC Cloud Native Services 8
Upgrading BRM REST Services Manager 10
Upgrading Your Business Operations Center Cloud Native Services 11
Upgrading Your Business Operations Center Cloud Native Service from 12.0.0.7.0 or Earlier to 16.26.0 12
Upgrading Your Business Operations Center Cloud Native Service from 12.0.0.8.0 to 16.26.0 14
Upgrading Your PCC Cloud Native Services 15
Upgrading Your PCC Cloud Native Services from 12.0.0.x.0 or Earlier to 15.0.0.0.0 15
Upgrading Your PCC Cloud Native Services from 15.0 to 16.26.0 16
Upgrading Your Billing Care and Billing Care REST API Cloud Native Services 18
Upgrading Your Billing Care and Billing Care REST API Cloud Native Services from 12.0.0.7.0 or Earlier to 16.26.0 18
Upgrading Your Billing Care and Billing Care REST API Cloud Native Services from 12.0.0.8.0 to 16.26.0 20

18 Performing Zero-Downtime Upgrades

Performing a Zero-Downtime Upgrade of BRM 1
Performing a Zero Downtime Upgrade of PDC 4

Copyright © 2019, 2026, Oracle and/or its affiliates.

Installing PDC 15.x on More Than Two Sites 6

19 Performing Zero-Downtime Upgrades of Disaster Recovery Cloud Native Systems

About the Zero-Downtime Upgrade of an Active-Active Disaster Recovery System 1
Tasks for Upgrading a BRM Cloud Native Active-Active System 2
Switching Off Site 2 3
Uninstalling BRM and ECE from Site 2 5
Upgrading Your BRM Database Schema in Site 2 6
Installing BRM 15.x Cloud Native on Site 2 8
Dropping the ECE Persistence Database Schema from Site 2 9
Installing ECE 15.x Cloud Native on Site 2 10
Failing Over Site 1 to Site 2 11
Uninstalling BRM and ECE from Site 1 15
Installing BRM Cloud Native on Site 1 16
Dropping the ECE Persistence Database Schema from Site 1 18
Installing ECE 15.x Cloud Native on Site 1 18
Federating ECE Cache Data Between Site 1 and Site 2 19

20 Migrating from On-Premise BRM to BRM Cloud Native

Migrating to BRM Cloud Native 1
Migrating from PDC On Premises to PDC Cloud Native 1

21 Rolling Back Your BRM Database Upgrade

About Rolling Back Your BRM Database 1
Tasks for the Database Rollback Process 2
About Reusing Your Previous BRM Installation 2

Part VI Troubleshooting BRM Cloud Native Deployments

22 Troubleshooting Your BRM Cloud Native Deployment

Problems with the Helm Installation 1
Helm Installation Fails with Time-Out Error 2
BRM Cloud Native Deployment Out of Memory Errors 3
PDC Messages Stuck in Rating Engine Queues 3
PDC Interceptor Pod is Started But Went to Error State 3
eceTopology.conf Errors While Restarting Pods 4
SyncPDC Pod Fails to Start and Enters CrashLoopBackOff State 5

Copyright © 2019, 2026, Oracle and/or its affiliates.

Billing Care Web Services Token Authentication Fails After Custom KeyStore Configuration 5
CM Fails During Deployment with Bad Login Error for /service/pcm_client 6

Copyright © 2019, 2026, Oracle and/or its affiliates.

About This Content

This guide describes how to install and administer Oracle Communications Billing and Revenue Management (BRM) Cloud Native Deployment Option.

Audience

This document is intended for those involved in installing and maintaining a BRM Cloud Native Deployment.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Part I

Overview of BRM Cloud Native

This part provides an overview of the Oracle Communications Billing and Revenue Management (BRM) cloud native deployment. It contains the following chapters:

- Overview of the BRM Cloud Native Deployment
- About the BRM Cloud Native Deployment Packages

Copyright © 2019, 2026, Oracle and/or its affiliates.

1

Overview of the BRM Cloud Native Deployment

Learn about configuring Oracle Communications Billing and Revenue Management (BRM) to run as a cloud native application in a containerized and orchestrated deployment architecture.

Topics in this document:

- About the BRM Cloud Native Deployment
- BRM Cloud Native Deployment Architecture

About the BRM Cloud Native Deployment

Oracle Communications Billing and Revenue Management (BRM), along with the following BRM applications, are available in a cloud native deployment option, supporting a Kubernetes-orchestrated containerized multi-service architecture to facilitate continuous integration, continuous delivery, and DevOps practices. This allows you to harness the benefits of the cloud with BRM's services.

- Oracle Communications Pricing Design Center (PDC)
- Oracle Communications Elastic Charging Engine (ECE)
- Oracle Communications Pipeline Configuration Center (PCC)
- Oracle Communications Collections Configuration Center
- Oracle Communications Billing Care
- Oracle Communications Business Operations Center

> **Note**
>
> You can also deploy Oracle Communications Offline Mediation Controller on a cloud native environment. See *Offline Mediation Controller Cloud Native Installation and Administration Guide* for more information.

You can set up your own BRM cloud native environment or build your own images of BRM and its applications. You use the cloud native deployment package to automate the deployment of BRM products and speed up the process to get services up and running, with product deployments preconfigured to communicate with each other through Helm charts.

BRM Cloud Native Deployment Architecture

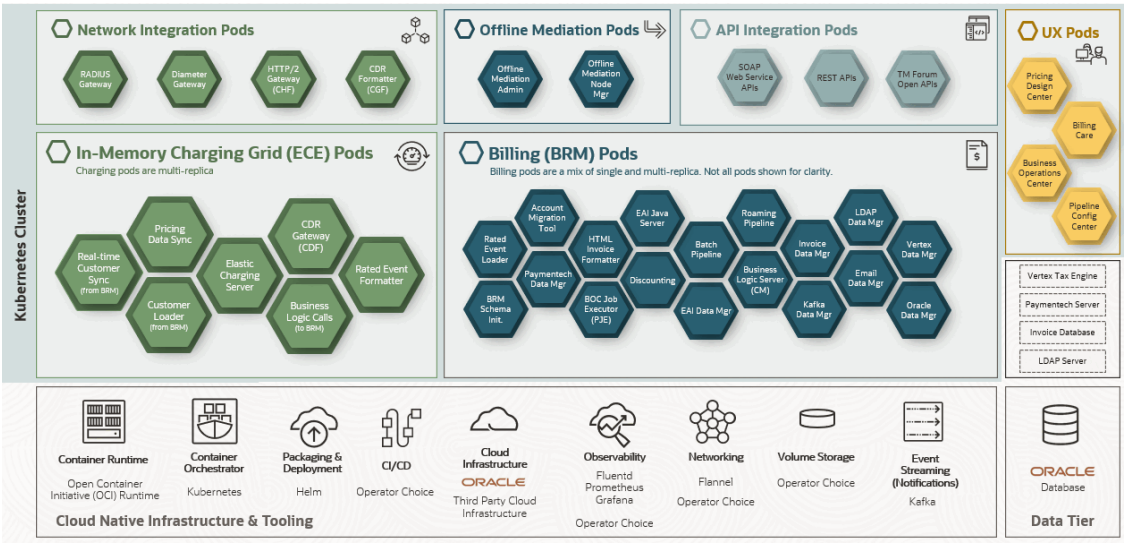
In the BRM cloud native architecture, each BRM service runs as a container and deploys as a Kubernetes pod, which is the fundamental building block of Kubernetes. Many core BRM services can be deployed and managed as multiple replicas within a Kubernetes replica set.

Figure 1-1 shows the pods and other components in a typical BRM cloud native deployment.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> Not all pods are shown for clarity. Pod names are descriptive and may differ from actual names in some cases.

Figure 1-1 BRM Cloud Native Deployment Architecture



In a cloud native deployment:

- PDC, Billing Care, Business Operations Center, Collections Configuration Center, and PCC are client applications. They connect to the CM, which represents the business logic layer of BRM, by using the Portal Communications Protocol (PCP).
- The CM communicates with other pods, which represent the data management layer of BRM, by using PCP.
- All PCP communication is encrypted using Transport-Layer Security (TLS).
- The data managers (DMs) interact with other downstream products that run the business logic.

  The downstream products can be containers or an on-premise system.
- ECE rates events and applies charges.
- Rating files for the batch pipeline are fed in through a Kubernetes PersistentVolumeClaim (PVC). The batch pipeline output is also available in a PVC for consumption by the Rated Event (RE) Loader pod.

The figure contains the following pods:

- A Kubernetes cluster, which sits in the middle of the architecture. It includes all Network Integration pods, Offline Mediation pods, API Integration pods, In-Memory Charging Grid (ECE) pods, Billing (BRM) pods, and User Experience (UX) pods.
  - Network Integration pods: This consists of the following:
    * RADIUS Gateway: This is a network-facing RADIUS Server.
    * Diameter Gateway: This is a network-facing Diameter Server.

Copyright © 2019, 2026, Oracle and/or its affiliates.

    - HTTP/2 (CHF) Gateway: This is an integrated Charging Function (CHF) capability that serves 5G standalone (SA) HTTP/2 Service Based Architecture network requests.
    - CDR Formatter (CGF): The call detail record (CDR) formatter extracts unrated 5G SA CDRs from the CDR storage database and formats them for the Kafka messaging service or disk storage. The CHF-CDR format is aligned with 3GPP TS 32.298 v16.5.0.
  - Offline Mediation pods: This consists of the following:
    - Offline Mediation Administrator: This is the Offline Mediation Controller administrator server.
    - Offline Mediation Node Manager: This is the Offline Mediation Controller node manager.
  - API Integration pods: This consists of the following:
    - SOAP Web Service APIs: This supports SOAP Web Services APIs.
    - REST APIs: This supports REST APIs.
    - TM Forum Open APIs: This supports TM Forum-aligned REST APIs.
  - In-Memory Charging Grid (ECE) pods: This consists of the following:
    - Real-time Customer Sync (from BRM): This applies synchronous or real-time customer account data updates from BRM to the in-memory charging grid.
    - Pricing Data Sync: This loads the pricing data from PDC into the charging grid.
    - Customer Loader (from BRM): This supports the initial loading and asynchronous updates of customer, data, credit limit data, offer profiles, product offering cross-reference data, and configuration objects from BRM to the Elastic Charging Engine.
    - Elastic Charging Server: This is the core in-memory grid-based charging service that supports rating, balance management, and session management.
    - CDR Gateway (CDF): This generates 5G SA unrated CDRs by processing requests from the HTTP/2 Gateway and persists them in the CDR storage database.
    - Business Logic Calls (to BRM): This sends information to the BRM server when business logic must be triggered or when updates are made to the BRM database.
    - Rated Event Formatter: This formats the rated events generated from the in-memory charging grid to BRM.
  - Billing (BRM) pods: This consists of the following:
    - Rated Event Loader: This loads the rated events generated by the in-memory charging grid into the BRM Database.
    - BRM Schema Initializer: This initializes the BRM Schema.
    - Account Migration Tool: This moves accounts between BRM schemas.
    - Paymentech Data Manager: This provides an interface to the Paymentech credit card processing service.
    - HTML Invoice Formatter: This is an HTML formatter for invoices.
    - BOC Job Executor (PJE): This processes batch jobs through Business Operations Center.

Copyright © 2019, 2026, Oracle and/or its affiliates.

    - EAI Java Server: This enables you to integrate BRM with other Java-based data-sharing applications in your system.
    - Discounting: This is the real-time discounting for subscription charging and billing.
    - Batch Pipeline: This is a rating engine used for rating large groups of offline charges.
    - EAI Data Manager: This supports the publishing of business events for external Enterprise Application Integration (EAI).
    - Roaming Pipeline: This processes roaming CDRs.
    - Business Logic Server (CM): This is an extensible and configurable business logic server that supports a rich set of revenue management functionality covering billing, invoicing, and accounts receivable processes.
    - Invoice Data Manager: This is a data manager for moving invoices to an external database.
    - Kafka Data Manager: This is a pod that supports integration to a Kafka messaging server for synchronizing internal business notification events to an external system.
    - LDAP Data Manager: This allows you to integrate BRM cloud native with an LDAP directory server. It replicates account and service data in the BRM database to the LDAP database.
    - Email Data Manager: This enables you to send email notifications and invoices to your customers.
    - Vertex Data Manager: This allows you to integrate BRM with the Vertex Q Series tax processing software.
    - Taxation Gateway: This allows you to integrate BRM with the Vertex O Series tax processing software.
    - Oracle Data Manager: This provides the integration between the core BRM business logic server and the Oracle Database. It performs Object Relational Mapping.
  - The User Experience pods: This consists of the following:
    - Pricing Design Center: This is an offer design tool and product catalog that supports charging and billing.
    - Billing Care: This is an agent that assists customer account management and a care application.
    - Collections Configuration Center: This is a web-based application that agents use to manage bill units that are past due and in collections.
    - Business Operations Center: This helps create, schedule, and view the revenue management-related business operations. This includes billing, invoicing, payment collection, refunds, B/L reporting, and custom jobs.
    - Pipeline Configuration Center: This is a web-based application that serves as the user interface for Pipeline Manager.

Copyright © 2019, 2026, Oracle and/or its affiliates.

2

About the BRM Cloud Native Deployment Packages

Learn about the Helm charts and images in the Oracle Communications Billing and Revenue Management (BRM) cloud native deployment package that help you deploy and manage pods of BRM product services in Kubernetes.

Topics in this document:

- Overview of the BRM Cloud Native Deployment Package
- About BRM Pods
- About Client Pods and Images
- About BRM PVCs and Pods

Overview of the BRM Cloud Native Deployment Package

The BRM cloud native deployment package includes the following:

- Ready-to-use images and Helm charts to help you orchestrate containers in Kubernetes.
- Sample Dockerfiles and scripts that you can use as a reference for building your own images.

You can use the images and Helm charts to help you deploy and manage pods of BRM product services in Kubernetes. Communication between pods of services of BRM products is preconfigured in the Helm charts.

About BRM Pods

Table 2-1 lists the pods for BRM whose containers are created and services are exposed through them.

Table 2-1 BRM Pods

Pod Name	Replica Type (1)	Container Port	Container Port Name	Service Type
amt	Single-replica	N/A	N/A	N/A
batch-controller	Multi-replica	N/A	N/A	N/A
batch-wireless-pipe	Single-replica	24001	batchpipe	ClusterIP
brm_apps_jobs	N/A	N/A	N/A	N/A
brmgateway	Multi-instance, 1 instance per schema	15502	brmgateway	ClusterIP
brm-sdk	Single-replica	N/A	N/A	N/A

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 2-1 (Cont.) BRM Pods

Pod Name	Replica Type (1)	Container Port	Container Port Name	Service Type
cm	Multi-replica	11960 (cm) 11961 (perflib, metrics) 11932 (eai-java-server, metrics)	cm-pcp-port cm-perflib-port eai-prom-port	ClusterIP
config_job	N/A	N/A	N/A	N/A
configloader	Single-replica	N/A	N/A	N/A
customerupdater	Multi-instance, 1 instance per schema	31022	N/A	N/A
diametergateway	Multi-instance, each instance can have 1 replica	3868	N/A	NodePort
dm-eai	Multi-replica	11970	dm-pcp-port	ClusterIP
dm-email	Multi-replica	17777	dm-pcp-port	ClusterIP
dm-fusa	Single-replica	15772	dm-fusa-port	ClusterIP
dm-invoice	Multi-replica	27777	dm-pcp-port	ClusterIP
dm-kafka	Multi-replica	12010 12012 (metrics)	dm-pcp-port dm-prom-port	ClusterIP
dm-ldap	Multi-replica	12850	dm-pcp-port	ClusterIP
dm-oracle	Multi-replica	12950 12951 (perflib, metrics)	dm-pcp-port dm-perflib-port	ClusterIP
dm-prov-telco	Multi-replica	20315	dm-pcp-port	ClusterIP
dm-vertex	Multi-replica	31274	dm-vertex-port	ClusterIP
ecs	Multi-replica	31022	N/A	N/A
emgateway	Multi-replica	15502	N/A	ClusterIP
event-stream-processor	Multi-replica	8080	esp-port	ClusterIP
formatter	Multi-replica	22272	formatter-port	ClusterIP
fusa-simulator	Single-replica	9780 (answer_s, online simulator) 8780 (answer_b, online simulator)	answer-s-port answer-b-port	ClusterIP
httpgateway	Multi-replica	8080	N/A	NodePort
init-db	Single-replica	N/A	N/A	N/A
pje	Multi-replica	31960	pje-pcp-port	ClusterIP
pricingupdater	Single-replica	9999	N/A	N/A
radiusgateway	Multi-replica	1812	N/A	NodePort
ratedeventformatter	Multi-instance, 1 instance required for each role on each schema (2)	9999	N/A	N/A
realtime-pipe	Multi-replica	24000	rtp	ClusterIP
rel-daemon	Multi-replica	N/A	N/A	N/A

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 2-1 (Cont.) BRM Pods

Pod Name	Replica Type (1)	Container Port	Container Port Name	Service Type
rel-job	Single-replica	N/A	N/A	N/A
rel-manager-job	Single-replica	N/A	N/A	N/A
rated-event-manager	Multi-instance, 1 instance per schema	8080	rem-metrics	N/A
roampipe	Single-replica	24002	roampipe	ClusterIP
taxation-gateway	Multi-replica	16025	tax-gw-port	ClusterIP

Note:

1. The BRM cloud native pods support the following replica types:
   - **Multi-replica**: You can scale these pods at the Kubernetes level to the number of replicas needed.
   - **Single-replica**: You cannot scale these pods.
   - **Multi-instance**: You can scale these pods but not at the Kubernetes level. Instead, a new instance needs to be created and scaling is usually limited.
2. The ratedeventformatter pod requires one instance for each role on each schema. For example, suppose you have two schemas on two sites. In that case, you create primary and secondary instances for each schema in site 1 and primary and secondary instances for each schema in site 2, for a total of 8 instances.

About Client Pods and Images

Table 2-2 lists the pods and images for PDC, PDC REST Services Manager, Pipeline Configuration Center, Billing Care, Business Operations Center, Collections Configuration Center, and BRM REST Services Manager.

> **Note**
>
> For the list of pods and images for Offline Mediation Controller, see *Offline Mediation Controller Cloud Native Installation and Administration Guide*.

Table 2-2 Client Pods and Images

Pod	Replica Type (1)	Image (2)	Container Port	Service Type	Access URL
bcws	Multi-replica (4)	**bcws:***tag*	7011 (admin-server) 8001 (managed-serverN and cluster-1) 8080 (monitoring-exporter, if monitoring is enabled)	ClusterIP	*host***:***port***/bcws**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 2-2 (Cont.) Client Pods and Images

Pod	Replica Type (1)	Image (2)	Container Port	Service Type	Access URL
billingcare	Multi-replica (4)	**billingcare:***tag*	7011 (admin-server) 8001 (managed-serverN and cluster-1) 8080 (monitoring-exporter, if monitoring is enabled)	ClusterIP	*host*:*port*/**bc**
boc	Multi-replica	**boc:***tag*	7011 (admin-server) 8001 (managed-serverN and cluster-1) 8080 (monitoring-exporter, if monitoring is enabled)	ClusterIP	*host*:*port*/**opsdashboard**
brm-rest-services-manager	Multi-replica	**brm-rest-services-manager:***tag*	9090 (HTTP) 8080 (HTTPS) 9060 (adminPort)	ClusterIP	*host*:*port*/**brm**
brm-wsm	Multi-replica	**brm_wsm:***tag*	8080 (HTTP) 8443 (HTTPS)	ClusterIP	*host*:*port*/**metro** *host*:*port*/**configurations/endpoints** *host*:*port*/**configurations/endpoints/default**
brm-collections-configuration-center-deployment	Multi-replica	**brm-collections-configuration-center-deployment:***tag*	32000 (HTTP) 32001 (HTTPS) 32080 (adminPort)	ClusterIP	*host*:*port*/**ccc**
pcc	Multi-replica	**pcc:***tag*	7012 (HTTPS)	ClusterIP	*host*:*port*/**pcc**
pdc	Single-replica	**pdc:***tag*	8001 (HTTP) (3)	ClusterIP	*host*:*port*/**pdc**
pdcrsm	Multi-replica	**pdcrsm:***tag*	8080	ClusterIP	*host*:*port*/**productCatalogManagement**
brmdomain	Multi-replica	**brm_wsm_wls:***tag*	7001 (admin-server) 8001 (managed-server*N* and cluster-1) 8080 (monitoring-exporter, metrics)	default default metrics	*host*:*port*/**BrmWebServices**
webhook	Single-replica	**webhook:***tag*	8080	ClusterIP	N/A

Note:

1. The BRM cloud native pods support the following replica types:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **Multi-replica**: You can scale these pods at the Kubernetes level to the number of replicas needed.
- **Single-replica**: You cannot scale these pods.
- **Multi-instance**: You can scale these pods but not at the Kubernetes level. Instead, a new instance needs to be created and scaling is usually limited.

2. Replace *tag* with the release version number, such as 16.26.0.0.0.0.
3. If the PDC user sets the **t3ChannelPort** and **t3sChannelPort** keys in the **values.yaml** file, the HTTP, HTTPS, t3Channel, and t3sChannel ports will be NodePort.
4. The Billing Care and Billing Care REST API pods support multi-replica managed services with scaling done through WebLogic Kubernetes Operator.

About BRM PVCs and Pods

Table 2-3 lists the PVCs and pods in a BRM cloud native deployment.

Table 2-3 List of PVCs in BRM Server

PVC Name	Pods
bcws-domain-domain-pvc	bcws-domain-deployer bcws-domain-admin-server bcws-domain-managed-server*N*
bcws-domain-batch-payment-pvc	bcws-domain-deployer bcws-domain-admin-server bcws-domain-managed-server*N*
billingcare-domain-domain-pvc	billingcare-domain-deployer billingcare-domain-admin-server billingcare-domain-managed-server*N*
billingcare-domain-batch-payment-pvc	billingcare-domain-deployer billingcare-domain-admin-server billingcare-domain-managed-server*N*
boc-domain-domain-pvc	boc-domain-deployer boc-domain-admin-server boc-domain-managed-server*N*
brm-sdk	brm-sdk
cmt-pvc	brm-apps-job pje
common-semaphore	batch-wireless-pipe realtime-pipe roampipe
ctqdir	dm-vertex
custom-job-file	brm-apps-job cm
data	batch-wireless-pipe roampipe
dm-kafka	dm-kafka

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 2-3 (Cont.) List of PVCs in BRM Server

PVC Name	Pods
fusa-temp	dm-fusa
oms-rel-archive	rel-daemon rel-job
oms-rel-data	rel-daemon rel-manager-job
oms-rel-input	rel-daemon rel-job
oms-rel-reject	rel-daemon rel-job
oms-rem-archive	rated-event-manager rel-manager-job
oms-rem-data	rated-event-manager rel-manager-job
oms-rem-input	rated-event-manager rel-manager-job
oms-rem-reject	rated-event-manager rel-manager-job
oms-uel-archive	batch-controller
oms-uel-input	batch-controller
oms-uel-reject	batch-controller
outputcdr	batch-wireless-pipe rel-daemon rel-job
outputreject	batch-wireless-pipe rel-daemon rel-job
pipelinelog	batch-wireless-pipe roampipe
pdc-app-pvc	PDC pod (PDC Application Container)
pdc-brm-pvc	PDC pod (PDC BRM Integration Pack)
roamoutputcdr	rel-daemon rel-job roampipe
roamoutputreject	rel-daemon rel-job roampipe
service-order	brm-apps-job dm-prov-telco
virtual-time	All pods

Table 2-4 lists the services associated with ECE.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 2-4 ECE Services

Service Name	Service Type	Port	Description
ece-brmgateway	ClusterIP	15502	BRM Gateway service
ece-cdrgateway	ClusterIP	8084	CDR Gateway service
ece-dgw	NodePort	3868	Diameter Gateway service
ece-emg	ClusterIP	External port	EM Gateway service
ece-jmx-service-external	NodePort	External port	JMX service
ece-http	NodePort	8080 31500	HTTP Gateway service
ece-monitoringagent-service-external	NodePort	External port	Monitoring agent service
ece-rgw	NodePort	1812	RADIUS Gateway service

For information about configuring these values, see "Configuring Gateway Service Types and NodePorts ."

Copyright © 2019, 2026, Oracle and/or its affiliates.

Part II

Getting Started with BRM Cloud Native Deployment

This part provides information about getting started with your Oracle Communications Billing and Revenue Management (BRM) cloud native deployment, including installing the prerequisite software and downloading the deployment package. It contains the following chapters:

- About Configuring and Deploying Your BRM Cloud Native Environment
- Setting Up Prerequisite Software
- Preparing Your BRM Cloud Native Environment

Copyright © 2019, 2026, Oracle and/or its affiliates.

3

About Configuring and Deploying Your BRM Cloud Native Environment

Learn about the high-level steps for configuring and deploying your Oracle Communications Billing and Revenue Management (BRM) cloud native environment.

Topics in this document:

- About Configuring and Deploying BRM Cloud Native
- High-Level Installation Tasks

About Configuring and Deploying BRM Cloud Native

You install the BRM cloud native deployment package by configuring and deploying its Helm charts. The Helm charts include YAML template descriptors for all Kubernetes resources and a **values.yaml** file that provides default configuration values for each chart.

Installing a Helm chart generates valid Kubernetes manifest files by replacing default values from the **values.yaml** file with custom values from your **override-values.yaml** file, and creates Kubernetes resources. Helm calls this a new release. You use the release name to track and maintain this installation.

> **Note**
>
> This documentation uses the **override-values.yaml** file name for ease of use, but you can name the file whatever you want.

The BRM cloud native deployment package includes the Helm charts in Table 3-1.

Table 3-1 BRM Cloud Native Helm Charts

Chart Name	Description	When to Use
oc-cn-init-db-helm-chart	This chart initializes and upgrades the database schema for the BRM server. • In initialize mode, it: – Creates users, tablespaces, tables, schemas, views, procedures, indexes, and other database objects needed by BRM Server – Loads seed data • In upgrade mode, it modifies the existing database schema to match the current release's data model.	Use this chart in initialize mode when preparing a new BRM setup and have an empty database schema. Use this chart in upgrade mode when upgrading your schema to the latest release.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-1 (Cont.) BRM Cloud Native Helm Charts

Chart Name	Description	When to Use
oc-cn-op-job-helm-chart	This chart does the following: • Creates WebLogic Server domains for PDC, Billing Care, the Billing Care REST API, and Business Operations Center. • Installs PDC, Billing Care, the Billing Care REST API, and Business Operations Center in their respective domains. • Populates persistent volumes with domain and application files for sharing between WebLogic Server runtimes. • Creates the following PDC groups: – **PricingDesignAdmin**: This group's users have administrative privileges on PDC. They can perform operations on all PDC UI screens, pricing components, and setup components. – **PricingAnalyst**: This group's users have administrative privileges for pricing components and view-only privileges for setup components. – **PricingReviewer**: This group's users have view-only privileges for all pricing and setup components.	If you want to use Billing Care, Business Operations Center, Pricing Design Center, or the Billing Care REST API, install this chart before you install **oc-cn-helm-chart**.
oc-cn-helm-chart	This chart does the following: • Deploys BRM server, PDC, PCC, and Collections Configuration Center. • Starts the WebLogic servers for Billing Care, the Billing Care REST API, and Business Operations Center. • Exposes web clients as services outside of the cluster. • Shares persistent volumes between its services through persistent volume claims.	Install this chart to use the services of the BRM server, PDC, PCC, Collections Configuration Center, Billing Care, Business Operations Center, or the Billing Care REST API.
oc-cn-ece-helm-chart	This chart does the following: • Deploys ECE and its services. • Sets up the connection with the BRM server and PDC. • Configures sharing of persistent volumes with the BRM server.	Install this chart to use ECE as your convergent charging solution.

High-Level Installation Tasks

You install BRM cloud native on your system by performing these high-level tasks:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Install all prerequisite software for your BRM cloud native environment.

   See "Setting Up Prerequisite Software".

2. Prepare your deployment environment by downloading the BRM cloud native deployment package, extracting the Helm charts, and loading the BRM component images.

   See "Preparing Your BRM Cloud Native Environment".

3. Configure and deploy the BRM database schema in your cloud native environment.

   See "Deploying the BRM Database Schema".

4. Configure the BRM cloud native services that you want to include in your system, including:

   - BRM server and PDC services. See "Configuring BRM Server and PDC Services".
   - BRM and PDC REST services. See "Configuring REST Services".
   - BRM client services such as Billing Care and Business Operations Center. See "Configuring the BRM Client Services".
   - ECE services. See "Configuring ECE Services".

5. Deploy the BRM cloud native services in your cloud native environment.

   See "Deploying BRM Cloud Native Services".

Copyright © 2019, 2026, Oracle and/or its affiliates.

4

Setting Up Prerequisite Software

Learn about prerequisite tasks to perform before installing the Oracle Communications Billing and Revenue Management (BRM) cloud native deployment package, such as installing Podman and Helm.

Topics in this document:

- BRM Cloud Native Prerequisite Tasks
- Software Compatibility
- Creating a Kubernetes Cluster
- Installing Podman
- Installing Helm
- Creating and Configuring Your BRM Database
- Installing an External Provisioner
- Installing WebLogic Kubernetes Operator
- Exposing BRM Services Outside the Kubernetes Cluster
- Setting Up ECE Cloud Native Ingress and Egress Flows

> ⚠ **Caution**
>
> Oracle does not provide support for any prerequisite third-party software installation or configuration. The customer must handle any installation or configuration issues related to non-Oracle prerequisite software.

BRM Cloud Native Prerequisite Tasks

As part of preparing your environment for BRM cloud native, you choose, install, and set up various components and services in ways that are best suited for your cloud native environment. The following shows the high-level prerequisite tasks for BRM cloud native:

1. Ensure you have downloaded the latest supported software that is compatible with BRM cloud native.
2. Create a Kubernetes cluster.
3. Install Podman and a container runtime supported by Kubernetes.
4. Install Helm.
5. Create and configure a BRM database.
6. Install and configure an external provisioner.
7. If you plan to deploy Pricing Design Center (PDC) or Business Operations Center on WebLogic:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Install and configure WebLogic Kubernetes Operator.
- Expose BRM services outside the Kubernetes cluster using an ingress controller.

8. If you plan to deploy Elastic Charging Engine (ECE), install and set up an ingress controller and an egress controller.
9. If you plan to deploy a client UI or REST API, install an Identity Provider (IdP) such as Oracle Identity Cloud Service or Oracle Access Management.
10. If you plan to integrate your BRM cloud native deployment with a Kafka Server, install the Apache Kafka software. See "Apache Kafka Quickstart" on the Apache Kafka website for installation instructions.
11. If you plan to integrate your BRM cloud native deployment with Oracle Analytics Publisher, install Oracle Analytics Publisher. See "Installing the Oracle Analytics Server Software" in *Oracle Analytics Installing and Configuring Oracle Analytics Server* for installation instructions.

> **Note**
>
> The Oracle Analytics Publisher software was previously named Oracle Business Intelligence (BI) Publisher.

Prepare your environment with these technologies installed, configured, and tuned for performance, networking, security, and high availability. Make sure backup nodes are available in case of system failure in any of the cluster's active nodes.

The following sections provide more information about the required components and services, the options you can choose from, and how you must set them up for your BRM cloud native environment.

Software Compatibility

To run, manage, and monitor your BRM cloud native deployment, ensure you use the latest versions of all compatible software. See *BRM Compatibility Matrix*.

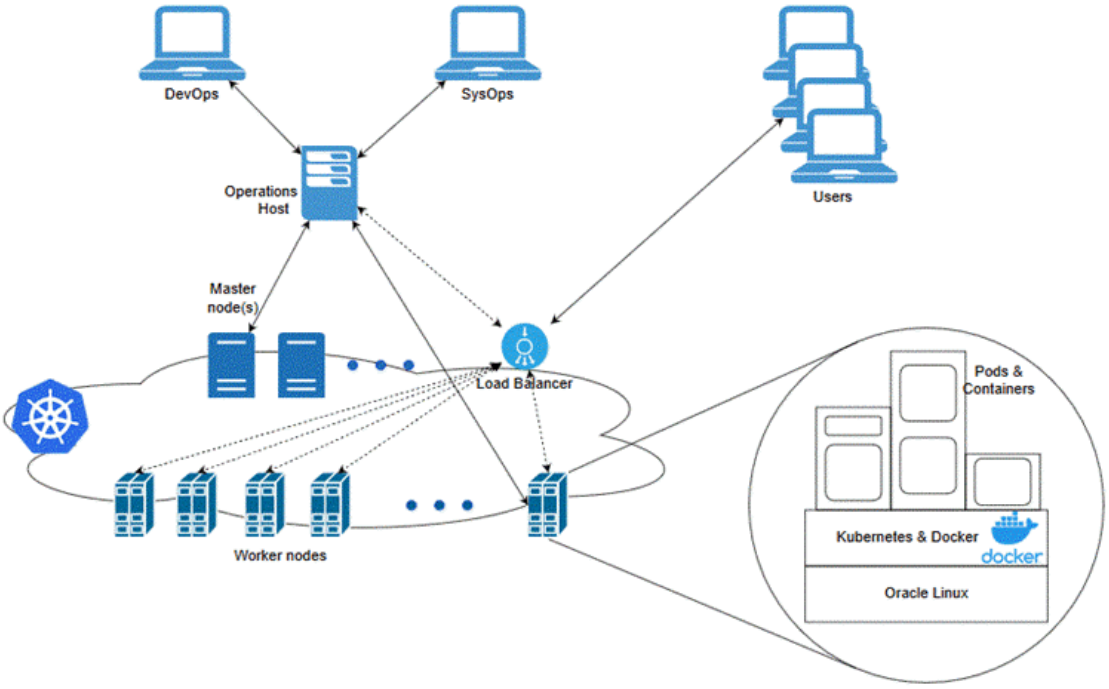
Creating a Kubernetes Cluster

Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. It groups containers into logical units for easier management and discovery. When you deploy Kubernetes, you get a physical cluster with machines called nodes. A reliable cluster must have multiple worker nodes spread over separate physical infrastructure, and a very reliable cluster must have multiple primary nodes spread over separate physical infrastructure.

Figure 4-1 illustrates the Kubernetes cluster and the components that it interacts with.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-1 Overview of the Kubernetes Cluster



Set up a Kubernetes cluster for your BRM cloud native deployment, securing access to the cluster and its objects with the help of service accounts and proper authentication and authorization modules. Also, set up the following in your cluster:

- **Volumes**: Volumes are directories that are accessible to the containers in a pod and provide a way to share data. The BRM cloud native deployment package uses persistent volumes for sharing data in and out of containers, but does not enforce any particular type. You can choose from the volume type options available in Kubernetes.
- **A networking model**: Kubernetes assumes that pods can communicate with other pods, regardless of which host they land on. Every pod gets its own IP address, so you do not need to explicitly create a link between pods or map container ports to host ports. Several implementations are available that meet the fundamental requirements of Kubernetes' networking model. Choose the networking model depending on the cluster requirement.

For more information about Kubernetes, see "Kubernetes Concepts" in the Kubernetes documentation.

Installing Podman

You use the Podman platform to containerize BRM products. Install Podman if you want to do one of these:

- Use the prebuilt images provided with the BRM cloud native deployment package.
- Build your own BRM images by writing your own Dockerfiles using the sample Dockerfiles from the BRM cloud native deployment package.

You can use Podman or any container runtime that supports the Open Container Initiative if it supports the Kubernetes version specified in *BRM Compatibility Matrix*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Installing Helm

Helm is a package manager that helps you install and maintain software on a Kubernetes system. In Helm, a package is called a *chart*, which consists of YAML files and templates rendered into Kubernetes manifest files. The BRM cloud native deployment package includes Helm charts that help create Kubernetes objects, such as ConfigMaps, Secrets, controller sets, and pods, with a single command.

The following shows sample steps for installing and validating Helm:

1. Download the Helm software from https://github.com/helm/helm/releases.

   For the list of supported Helm versions, see *BRM Compatibility Matrix*.

2. Extract the Helm files from the archive:

   ```
   tar -zxvf helm-version-linux-amd64.tar.gz
   ```

   where *version* is the Helm version number.

3. Find the helm binary in the unpacked directory and move it to your desired directory. For example:

   ```
   mv linux-amd64/helm /usr/local/bin/helm
   ```

4. Check the version of Helm:

   ```
   helm version
   ```

Helm uses a configuration file to allow the **helm** command to access the Kubernetes cluster. By default, this file is **$HOME/.kube/config**, but you can specify another location by setting it in the **$KUBECONFIG** environment variable. Helm inherits the permissions set up for this access into the cluster. If role-based access control (RBAC) is configured, you must grant Helm users sufficient cluster permissions.

For more information about installing Helm, see "Installing Helm" in the Helm documentation.

Creating and Configuring Your BRM Database

You must install an Oracle database accessible through the Kubernetes network so BRM cloud native pods can perform database operations. The Oracle database you use can be:

- On-premises, which can be either physical or VM
- Cloud-based, such as Bare Metal, VM, or DBaaS on Oracle Cloud Infrastructure

You can use an existing BRM database or create a new one. See *BRM Compatibility Matrix* for the latest supported database versions.

To create and configure a new BRM database:

1. When you install the Oracle database software, pay particular attention to the following requirements:
   - Install **Oracle Enterprise Edition**
   - Choose a **Customized** installation. This option lets you configure Oracle with the **AL32UTF8** database character set.
   - To configure discounts in BRM, install the following Oracle components:
     - **Oracle XML DB**. For more information, see the Oracle documentation.

Copyright © 2019, 2026, Oracle and/or its affiliates.

  - **Oracle XML Developer's Kit (XDK)**. For more information, see the Oracle documentation.

- Install Oracle JServer as part of the Oracle Database installation.
- To partition the tables in your BRM database, install the **Oracle Partitioning** component. See "Partitioning Tables" in *BRM System Administrator's Guide*.

2. When you create your BRM database, pay particular attention to the following requirements:
   - Specify a **Global Database Name** using the format *DatabaseName***.***DomainName*, where *DatabaseName* is the database name and *DomainName* is the network domain in which the database is located. For example, **pindb***hostname***.example.com**. Most BRM databases use a *DatabaseName* of **pindb***hostname*, but you can use another name.

     > **Note**
     >
     > You can modify your machine's default domain name in the **$ORACLE_HOME/network/admin/sqlnet.ora** file. For information, see your Oracle documentation.

   - Specify a **System Identifier (SID)** for your database. For clarity, it should be the same as your Oracle database name. Most BRM databases are named **pindb***hostname*, but you can use another name.
     - Set the **Character Set** to **AL32UTF8**.
     - Set the **National Character Set** to **AL16UTF16**.
3. Set your LD_LIBRARY_PATH environment variable to **$ORACLE_HOME/lib**.
4. (Optional) Set up TLS authentication in the BRM database. See "Configuring Transport Layer Security Encryption" in *Oracle AI Database Security Guide*. Also, ensure that you:
   - Create a TLS certificate or obtain one from a certificate provider
   - Install the certificate in the Oracle Database Server
5. Configure your database manually or let the **oc-cn-init-db-helm-chart** Helm chart configure the database for you.

   > **Note**
   >
   > For production systems, your database administrator must create the database manually.

   You can do one of the following:

   - **Use the oc-cn-init-db-helm-chart Helm chart to configure a demonstration database for you**

     If you provide the system administrator's user name and password, the init-db Helm chart can automatically configure your database for demonstration or development systems. For more information, see "Using the BRM Installer to Configure Your Database for Demonstration Systems" in *BRM Installation Guide*.

   - **Configure a demonstration database manually**

Copyright © 2019, 2026, Oracle and/or its affiliates.

You can configure your database manually so it contains additional or larger tablespaces. For more information, see "Configuring Your Database Manually for Demonstration Systems" in *BRM Installation Guide*.

- **Configure a production database manually**

  For production systems, your database administrator must create the tablespaces for the BRM data and indexes. For information on estimating your database size and creating tablespaces, see "Planning Your Database Configuration" in *BRM Installation Guide*. To set the default storage model values for your cloud native installation, set the values in the **configmap_partition_cfg.yaml** file instead of the **pin_tables.values** file.

6. Grant the BRM schema user select permission on the V$SESSION database table. To do so, connect to the Oracle database with SQL*Plus as the **system** user and then enter this command:

   ```
   SQL> GRANT SELECT ON TABLE V$SESSION TO brmSchemaUser;
   ```

The installers for PDC, Billing Care, and all other products automatically create the tablespaces and users that are required for those products.

Installing an External Provisioner

An external provisioner creates shared, persistent storage for containers in your BRM cloud native environment. Use shared persistent storage only for data that must be shared across pods or preserved outside the lifecycle of a pod, such as:

- Input data, such as pricing XML files
- Output data, such as archive files and reject files from Rated Event Loader and Universal Event Loader
- Data that needs to be shared between containers, such as **pin_virtual_time**

Install and set up an external provisioner that supports the access modes required by the volumes in your deployment. Use ReadWriteMany access for volumes that must be shared across pods. Some BRM cloud native components use ReadWriteOnce or pod-local storage where supported and do not require shared ReadWriteMany storage.

Installing WebLogic Kubernetes Operator

Oracle WebLogic Kubernetes Operator helps you to deploy and manage WebLogic domains in your Kubernetes environment. It consists of several parts:

- The operator runtime
- The model for a Kubernetes custom resource definition (CRD)
- A Helm chart for installing the operator

In the BRM cloud native environment, you use WebLogic Kubernetes Operator to maintain the domains and services for Billing Care, the Billing Care REST API, PDC, and Business Operations Center.

The following shows sample steps for installing WebLogic Kubernetes Operator on your BRM cloud native environment:

1. Add the Helm repository for WebLogic Kubernetes Operator:

   ```
   helm repo add weblogic-operator https://oracle.github.io/weblogic-kubernetes-operator/charts
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Create a new namespace for WebLogic Kubernetes Operator. For example, this **kubectl** command creates the namespace **operator**:

   ```
   kubectl create namespace operator
   ```

3. Install WebLogic Kubernetes Operator:

   ```
   helm install weblogic-operator weblogic-operator/weblogic-operator --namespace
   operator --version version
   ```

   where *version* is the version of WebLogic Kubernetes Operator, such as **4.2.13**. See *BRM Compatibility Matrix* for a list of supported versions.

   If the installation is successful, you will see something similar to this:

   ```
   NAME: weblogic-operator
   LAST DEPLOYED: Tue Oct 6 08:29:03 2025
   NAMESPACE: weblogic-operator
   STATUS: deployed
   REVISION: 1
   TEST SUITE: None
   ```

4. Check the pod:

   ```
   kubectl get pods
   ```

   You should see something similar to this:

   ```
   NAME                                 READY   STATUS    RESTARTS   AGE
   weblogic-operator-849cc6bdd8-vkx7n   1/1     Running   0          57s
   ```

For more information about WebLogic Kubernetes Operator, see "Introduction" in the WebLogic Kubernetes Operator documentation.

Exposing BRM Services Outside the Kubernetes Cluster

Using an ingress controller exposes BRM services outside the Kubernetes cluster and allows clients to communicate with BRM.

The ingress controller monitors the ingress objects and acts on the configuration embedded in these objects to expose BRM HTTP and T3 services to the external network. Adding an external load balancer provides highly reliable single-point access to the services exposed by the Kubernetes cluster. In this case, the ingress controller exposes the services on behalf of the BRM cloud native instance. Using a load balancer removes the need to expose Kubernetes node IPs to the larger user base, insulates users from changes (in terms of nodes appearing or being decommissioned) to the Kubernetes cluster, and enforces access policies.

If you are using Billing Care, the Billing Care REST API, or Business Operations Center, you must add a load balancer to your BRM cloud native system that has:

- Path-based routing for the WebLogic Cluster service.
- Sticky sessions enabled. That is, if the load balancer redirects a client's login request to Managed Server 1, all subsequent requests from that client are redirected to Managed Server 1.
- TLS enabled between the client and the load balancer to secure communications outside of the Kubernetes cluster.

  Business Operations Center and Billing Care use HTTP and rely on the load balancer to terminate HTTPS.

Copyright © 2019, 2026, Oracle and/or its affiliates.

For information about configuring an external load balancer for CM SSL/TLS connections, see "Configuring an External Load Balancer for Connection Manager SSL/TLS Connections" in *BRM Cloud Native System Administrator's Guide*.

See "Ingress" in the WebLogic Kubernetes Operator documentation for more information about setting up an ingress controller and sample load balancers.

Setting Up ECE Cloud Native Ingress and Egress Flows

Ingress and egress controllers expose ECE services outside the Kubernetes cluster, allowing external networks to communicate with ECE. For example, an ingress controller can route requests from Diameter and 5G HTTP clients to the httpgateway and diametergateway pods for processing. Likewise, an egress controller can send CDR records from the cdrformatter pod to the ECE database.

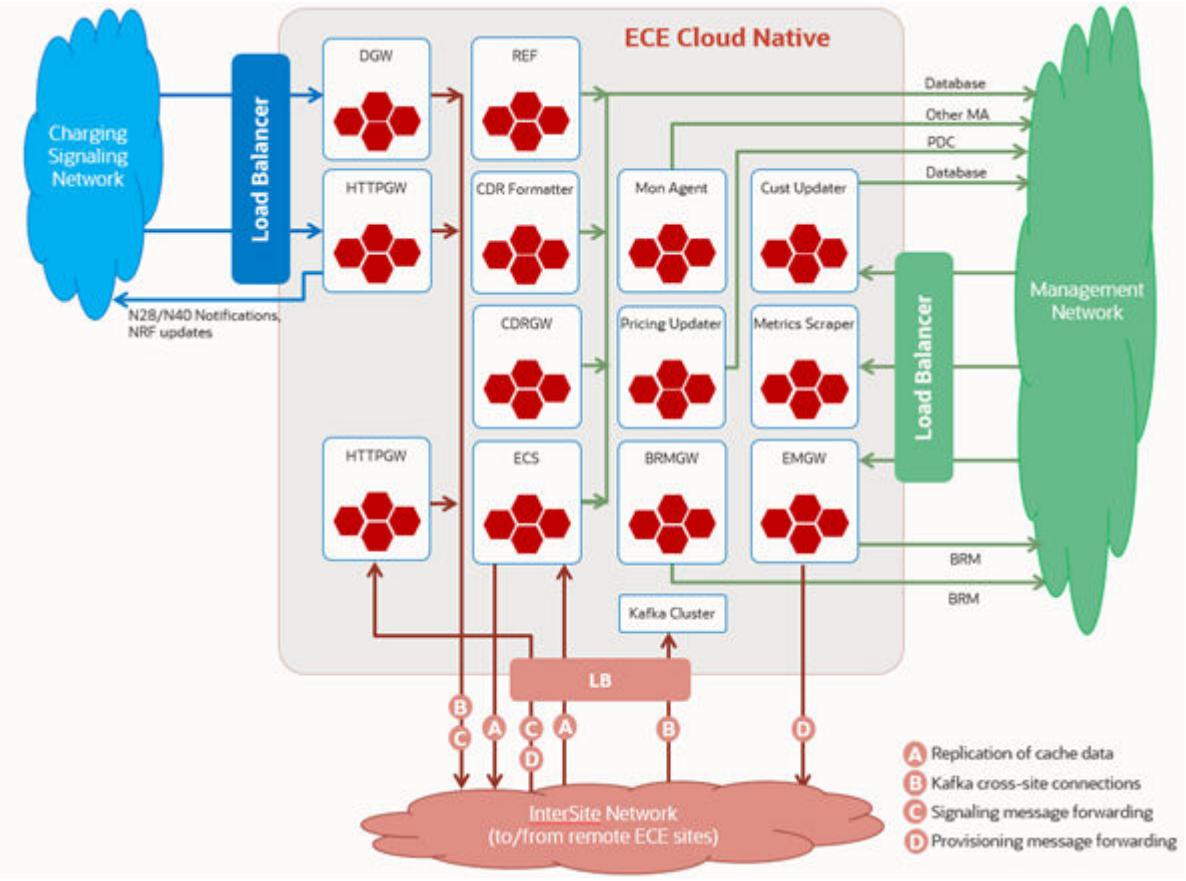
You can expose external network IPs for ingress traffic from external clients using the following:

- A load balancer exposing the IPs on the external network. The load balancer sends ingress traffic to Kubernetes services or node ports.
- Kubernetes service IPs or worker node IPs residing on the external network.

You can route egress traffic through an external network IP that is hosted by a worker node interface.

Figure 4-2 shows an ECE cloud native deployment with sample ingress and egress flows.

Figure 4-2 ECE Cloud Native Ingress and Egress Flows



Copyright © 2019, 2026, Oracle and/or its affiliates.

In this figure:

- The ingress flows traverse a load balancer, but you can use an alternate ingress flow to meet your business requirements.
- The egress flows are depicted generically. Network source addressing and routing may vary based on your business requirements.
- ECE cloud native uses logical external networks. The number and content of these networks may vary depending on your business requirements.

Table 4-1 describes the egress flow from each ECE pod to an endpoint.

Table 4-1 ECE Cloud Native Egress Flows

ECE Pod	Egress Endpoints
brmgateway	ECE database BRM database **Note:** The BRM database is accessed during installation.
cdrformatter	ECE database **Note:** The ECE database is used for CDR management.
cdrgateway	ECE database **Note:** The ECE database is used for CDR management.
configloader	ECE database
customerupdater	ECE database BRM database
diametergateway	Remote HTTP Gateway (for active-active deployments only) Remote Kafka server (for active-active deployments only) **Note:** This pod does not initiate Diameter connections to Diameter signaling clients.
ece-customerloader-job	ECE database BRM database
ece-persistence-job	ECE database
ece-persistence-upgrade-job	ECE database
ecs	ECE database BRM database Remote ECE Coherence Federation (for active-active and active-standby deployments only) **Note:** The BRM database is accessed during customer loading.
emgateway	BRM database Remote HTTP Gateway (for active-active deployments only) **Note:** This pod forwards requests to the remote HTTP Gateway. This is optional for active-active deployments.
httpgateway	Charging signaling clients Remote HTTP Gateway (for active-active deployments only) Remote Kafka server (for active-active deployments only) **Note:** This pod sends HTTP/2 requests to 5G clients. **Note:** The httpgateway pod's egress to a remote ECE HTTP Gateway is needed only if it is processing 5G charging traffic.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 (Cont.) ECE Cloud Native Egress Flows

ECE Pod	Egress Endpoints
monitoringagent	Monitoring Agents Remote Monitoring Agent (for active-active and active-standby deployments only)
pricingupdater	Pricing Design Center
ratedeventformatter	ECE database BRM database (for Rated Event Manager plug-in only) **Note:** Direct access to the BRM database occurs only when the Rated Event Manager plug-in is configured to write rated events directly to the BRM database.

Copyright © 2019, 2026, Oracle and/or its affiliates.

5

Preparing Your BRM Cloud Native Environment

Learn how to prepare your system for the Oracle Communications Billing and Revenue Management (BRM) cloud native deployment by downloading the BRM cloud native Helm charts and BRM images.

Topics in this document:

- Tasks for Preparing Your BRM Cloud Native Environment
- Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files
- Pulling BRM Images from the Oracle Container Registry
- Downloading BRM Images from Oracle Software Delivery Website
- Pulling WebLogic Images for PDC, Billing Care, Billing Care REST API, and Business Operations Center

Tasks for Preparing Your BRM Cloud Native Environment

Prepare your system for the BRM cloud native deployment by performing the following high-level tasks:

1. Downloading the Helm charts for the BRM cloud native deployment. See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Downloading the BRM cloud native images in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. If you plan to deploy Pricing Design Center (PDC), Billing Care, the Billing Care REST API, or Business Operations Center, downloading the Oracle WebLogic cloud native image. See "Pulling WebLogic Images for PDC, Billing Care, Billing Care REST API, and Business Operations Center".

Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files

To download the BRM cloud native Helm charts and Docker files:

1. Go to https://edelivery.oracle.com.
2. Sign in to the Oracle Software Delivery website using an Oracle account.
3. Search for and select the following releases and then click **Continue**.
   - Oracle Communications Billing and Revenue Management Cloud Native Deployment Option 16.26.0.*x*.*x*

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Oracle Communications Elastic Charging Engine Cloud Native Deployment Option 16.26.0.*x*.*x*
- Oracle Communications Pricing Design Center Cloud Native Deployment Option 16.26.0.*x*.*x*

4. Select the following in the download queue and click **Continue**:
   - Oracle Communications Billing and Revenue Management Cloud Native Deployment Option 16.26.0.*x*.*x*-CN
   - Oracle Communications Elastic Charging Engine Cloud Native Deployment Option 16.26.0.*x*.*x*-CN
   - Oracle Communications Pricing Design Center Cloud Native Deployment Option 16.26.0.*x*.*x*-CN
5. Accept the Oracle standard terms and restrictions and then click **Continue**.
6. Select the following packages and then click **Download**:
   - Oracle Communications Cloud Native Helm Chart 16.26.0.*x*.*x*
   - Oracle Communications Elastic Charging Engine Cloud Native Deployment Option Helm Chart 16.26.0.*x*.*x*
   - Oracle Communications Cloud Native Operator Job Helm Chart 16.26.0.*x*.*x*
   - Oracle Communications Cloud Native Database Initializer Helm Chart 16.26.0.*x*.*x*
   - Oracle Communications Cloud Native Docker Build Files 16.26.0.*x*.*x*
   - Oracle Communications Elastic Charging Engine Cloud Native Docker Files 16.26.0.*x*.*x*
   - Oracle Communications Cloud Native Pricing Design Center 16.26.0.*x*.*x*

   Each package is downloaded to a separate Zip file.
7. Extract the following Helm chart and Docker archive files from each Zip file:
   - BRM Helm Chart: **oc-cn-helm-chart-16.26.0.*x*.*x*.tgz**
   - ECE Helm Chart: **oc-cn-ece-helm-chart-16.26.0.*x*.*x*.tgz**
   - ECE Composable Services Helm Chart: **oc-ccs-16.26.0.0.0+98a3cbb6.tgz**
   - Operator Job Helm Chart: **oc-cn-op-job-helm-chart-16.26.0.*x*.*x*.tgz**
   - Database Initializer Helm Chart: **oc-cn-init-db-helm-chart-16.26.0.*x*.*x*.tgz**
   - BRM Dockerfiles: **oc-cn-docker-files-16.26.0.*x*.*x*.tgz**
   - ECE Dockerfiles: **oc-cn-ece-docker-files-16.26.0.*x*.*x*.tgz**
8. Extract the Helm charts and Dockerfiles from the archive files by running these commands:

```
tar xvzf oc-cn-helm-chart-16.26.0.x.x.tgz

tar xvzf oc-cn-ece-helm-chart-16.26.0.x.x.tgz

tar xvzf oc-ccs-16.26.0.0.0+98a3cbb6.tgz

tar xvzf oc-cn-op-job-helm-chart-16.26.0.x.x.tgz

tar xvzf oc-cn-init-db-helm-chart-16.26.0.x.x.tgz

tar xvzf oc-cn-docker-files-16.26.0.x.x.tgz

tar xvzf oc-cn-ece-docker-files-16.26.0.x.x.tgz
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 5-1 lists the files and directories extracted from the archive files.

Table 5-1 Extracted Files

Archive File	Extracted Directories
oc-cn-helm-chart-16.26.0.*x*.*x*.tgz	**oc-cn-helm-chart** directory: Contains the BRM Helm chart files. **sample_configurations** directory: This directory contains the default configuration XML files, such as **bus_params_AR.xml** and **pin_config_export_gl.xml**.
oc-cn-ece-helm-chart-16.26.0.*x*.*x*.tgz	**oc-cn-ece-helm-chart** directory: Contains the ECE Helm chart files.
oc-ccs-16.26.0.0.0+98a3cbb6.tgz	**oc-ccs** directory: Contains the ECE composable services Helm chart files.
oc-cn-op-job-helm-chart-16.26.0.*x*.*x*.tgz	**oc-cn-op-job-helm-chart** directory: Contains the WebLogic Operator Job Helm chart files.
oc-cn-init-db-helm-chart-16.26.0.*x*.*x*.tgz	**oc-cn-init-db-helm-chart** directory: Contains the Database Initializer Helm chart files.
oc-cn-docker-files-16.26.0.*x*.*x*.tgz	**oc-cn-docker-files** directory: Contains the Dockerfiles for BRM, PDC, PDC REST Services Manager, Collections Configuration Center, Pipeline Configuration Center, Business Operations Center, and Billing Care.
oc-cn-ece-docker-files-16.26.0.*x*.*x*.tgz	**docker_files** directory: Contains the Dockerfiles for ECE.

Pulling BRM Images from the Oracle Container Registry

To pull BRM cloud native images, such as the Connection Manager (CM) image and the Data Manager (DM) image, from the Oracle Container Registry, do the following:

1. In a web browser, go to https://container-registry.oracle.com.
2. Sign in to the Oracle Container Registry using an Oracle account.

> **Note**
>
> To pull images for licensed software on the Oracle Container Registry, you must have an Oracle account. You can create an Oracle account at https://profile.oracle.com/myprofile/account/create-account.jspx.

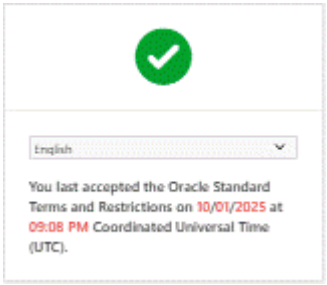
3. Select the **Oracle Communications Cloud Scale Monetization** container.

   The Oracle Communications Cloud Scale Monetization page appears.
4. Select one of the repository names from Table 5-2.

   The repository page appears.
5. Accept the Oracle terms and restrictions by:
   a. (For non-CPU images) Selecting your desired language.
   b. Clicking **Continue**.
   c. Scrolling to the bottom of the terms and restrictions pages and clicking **Accept**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

If successful, you will see something similar to this:



6. On your host system, log in to the Oracle Container Registry using the Podman command-line interface (CLI):

   ```
   podman login container-registry.oracle.com
   ```

7. When prompted for a user name and password, enter your Oracle credentials.
8. Pull the BRM cloud native image from the registry:

   ```
   podman pull container-registry.oracle.com/communications_monetization/imageName:tag
   ```

   where:

   - *imageName* is the name of a software image listed in Table 5-2.
   - *tag* is the tag name for the BRM cloud native image, such as **16.26.0.***x.x*.

   For example, to pull the CM cloud native image from the registry:

   ```
   podman pull container-registry.oracle.com/communications_monetization/cm:16.26.0.x.x
   ```

   The image is pulled from the Oracle Container Registry and stored locally, where it is ready to be used to deploy containers.

9. Confirm the images have been pulled from the Oracle Container Registry:

   ```
   podman images
   ```

   If successful, you will see something similar to this:

   ```
   REPOSITORY
   TAG            IMAGE ID        CREATED
   container-registry.oracle.com/communications_monetization/cm
   16.26.0.x.x        133dd3580b87    2 seconds ago
   container-registry.oracle.com/communications_monetization/dm_kafka
   16.26.0.x.x        136dd3593b47    3 seconds ago
   ```

10. Log out of the registry to prevent unauthorized access and to remove any record of sign-in credentials that Podman might store for future operations:

    ```
    podman logout container-registry.oracle.com
    ```

Table 5-2 lists the image names for the BRM cloud native components.

Table 5-2 BRM Cloud Native Images

Component Name	Image Name
Batch Controller	**batch_controller**
Batch Pipeline	**batch_pipeline**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 5-2 (Cont.) BRM Cloud Native Images

Component Name	Image Name
Billing Care	**oracle/billingcare**
Billing Care REST API	**oracle/bcws**
BRM Applications	**brm_apps**
BRM REST Services Manager	**oracle/brm-rest-services-manager**
BRM SDK	**brm_sdk**
Business Operations Center	**oracle/boc**
Connection Manager	**cm**
Collections Configuration Center	**oracle/brm-collections-configuration-center**
Database Initializer	**init_db**
Database Upgrade	**upgrade**
Elastic Charging Engine	**oc-cn-ece**
Email Data Manager	**dm_email**
Enterprise Application Integration (EAI) Data Manager	**dm_eai**
EAI Java Server	**eai_js**
Event Stream Processor	**event_stream_processor**
Invoice Data Manager	**dm_invoice**
Invoice Formatter	**formatter**
Kafka Data Manager	**dm_kafka**
LDAP Data Manager	**dm_ldap**
Oracle Data Manager	**dm_oracle**
Paymentech Data Manager	**dm_fusa**
Paymentech Simulator	**answer**
Performance Libraries	**perflib**
Pipeline Configuration Center	**oracle/pcc**
Pricing Design Center	**oracle/pdc**
Pricing Design Center REST Services Manager	**oracle/pdcrsm**
Provisioning Data Manager	**dm_prov_telco**
Rated Event Loader	**rel_daemon**
Rated Event Manager	**rem**
Real-Time Pipeline	**realtimepipe**
Roaming Manager	**roam_pipeline**
Taxation Gateway	**taxation_gateway**
Vertex Data Manager	**dm_vertex**
Webhook	**webhook**
Web Services Manager	**brm_wsm**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Downloading BRM Images from Oracle Software Delivery Website

To download BRM cloud native images, such as the Billing Care image, from the Oracle Software Delivery website:

1. Go to https://edelivery.oracle.com.
2. Sign in to the Oracle Software Delivery website using an Oracle account.
3. Search for and select **Oracle Communications Billing and Revenue Management Cloud Native Deployment Option 16.26.0.*x.x*** and then click **Continue**.
4. Select the following and then click **Continue**:
   - Oracle Communications Billing and Revenue Management Cloud Native Deployment Option 16.26.0.*x.x*-CN
   - Oracle Communications Pricing Design Center Cloud Native Deployment Option 16.26.0.*x.x*-CN
   - Oracle Communications Elastic Charging Engine Cloud Native Deployment Option 16.26.0.*x.x*-CN
5. Accept the Oracle standard terms and restrictions and then click **Continue**.
6. Select the packages listed in Table 5-3 and then click **Download**.

   Each package is downloaded to a separate Zip file.
7. Extract the package files listed in Table 5-3 from each Zip file.

> **Note**
>
> ECE includes packages for ECE composable services.
>
> If you use HTTP Gateway, CDR Gateway, and CDR Formatter to process 5G traffic, together with BRMCdrPlugin or the Rated Event Manager plug-in to process rated events, you do not need to install the following packages for your existing deployment. Install these packages only if you plan to migrate to ECE composable services:
>
> - **oc-ccs-16.26.0.*x.x*+98a3cbb6.tgz**
> - **oc-ccs-nrf-management-16.26.0.*x.x*.tar**
> - **oc-ccs-charging-gateway-16.26.0.*x.x*.tar**
> - **oc-ccs-nchf-converged-charging-16.26.0.*x.x*.tar**
> - **oc-ccs-db-client-16.26.0.*x.x*.tar**

Table 5-3 BRM Cloud Native Packages and Package Files

BRM Package Name	Package File Name
Oracle Communications Cloud Native Batch Controller	**oc-cn-brm-batch-controller-16.26.0.*x.x*.tar**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 5-3 (Cont.) BRM Cloud Native Packages and Package Files

BRM Package Name	Package File Name
Oracle Communications Cloud Native Batch Pipeline	**oc-cn-brm-batch-pipeline-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Billing Care	**oc-cn-billingcare-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Billing Care REST API	**oc-cn-bcws-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native BRM Applications	**oc-cn-brm-apps-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native BRM Database Initializer	**oc-cn-brm-init-db-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native BRM REST Services Manager	**oc-cn-brm-rest-services-manager-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native BRM SDK	**oc-cn-brm-sdk-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native BRM Taxation Gateway	**oc-cn-brm-taxation-gateway-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Business Operations Center	**oc-cn-boc-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Collections Configuration UI	**oc-cn-brm-collections-configuration-center-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Connection Manager	**oc-cn-brm-cm-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Database Upgrade	**oc-cn-brm-upgrade-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Elastic Charging Engine Cloud Native Deployment Option	**oc-cn-ece-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native NRF Management	**oc-ccs-nrf-management-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Charging Gateway Function	**oc-ccs-charging-gateway-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Charging Manager	**oc-ccs-nchf-converged-charging-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Database Client	**oc-ccs-db-client-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Email Data Manager	**oc-cn-brm-dm-email-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Enterprise Application Integration Data Manager	**oc-cn-brm-dm-eai-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Enterprise Application Integration Java Server	**oc-cn-brm-eai-js-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Event Stream Processor	**oc-cn-brm-esp-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Fusa Data Manager	**oc-cn-brm-dm-fusa-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Fusa Simulator	**oc-cn-brm-fusa-simulator-16.26.0.*x.x*.tar**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 5-3 (Cont.) BRM Cloud Native Packages and Package Files

BRM Package Name	Package File Name
Oracle Communications Cloud Native Invoice Data Manager	**oc-cn-brm-dm-invoice-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Invoice Formatter	**oc-cn-brm-invoice-formatter-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Kafka Data Manager	**oc-cn-brm-dm-kafka-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native LDAP Data Manager	**oc-cn-brm-dm-ldap-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Oracle Database Manager	**oc-cn-brm-dm-oracle-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Performance Profiling Toolkit	**oc-cn-brm-perflib-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Pipeline Configuration Center	**oc-cn-pcc-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Pricing Design Center	**oc-cn-pdc-16.26.0.*x.x*.tar** **oc-cn-pdc-rsm-16.26.0.*x.x*.tar** **oc-cn-pdc-rsm-jars-16.26.0.*x.x*.tar.gz**
Oracle Communications Cloud Native Provisioning Data Manager	**oc-cn-brm-dm-prov-telco-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Rated Event Loader	**oc-cn-brm-rel-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Rated Event Manager	**oc-cn-brm-rem-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Real-Time Pipeline	**oc-cn-brm-realtime-pipeline-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Roaming Manager	**oc-cn-brm-roam-pipeline-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Vertex Data Manager	**oc-cn-brm-dm-vertex-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Webhook	**oc-cn-brm-webhook-16.26.0.*x.x*.tar**
Oracle Communications Cloud Native Web Services Manager	**oc-cn-brm-wsm-16.26.0.*x.x*.tar**

8. Load each package file as an image into the Podman system using the following command:

   ```
   podman load --input fileName
   ```

   where *fileName* is the package file name listed in Table 5-3.

   For example, to load the Kafka DM image in the Podman system, enter this command:

   ```
   podman load --input oc-cn-brm-dm-kafka-16.26.0.x.x.tar
   ```

If you use an internal registry to access images from different Kubernetes nodes, push the images from your local system to the registry server. For example, if the registry is identified by *RepoHost*:*RepoPort*, you'd push the Kafka DM image to the registry using the Podman CLI like this:

1. Tag the Kafka DM image with the registry server:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
podman tag dm-kafka:16.26.0.x.x RepoHost:RepoPort/dm-kafka:16.26.0.x.x
```

2. Push the Kafka DM image to the registry server:

```
podman push RepoHost:RepoPort/dm-kafka:16.26.0.x.x
```

Pulling WebLogic Images for PDC, Billing Care, Billing Care REST API, and Business Operations Center

If you use PDC, Billing Care, Billing Care REST API, or Business Operations Center, pull the Oracle WebLogic image from the Oracle Container Registry into your private repository.

To load the Oracle WebLogic image into your private repository:

1. In a web browser, go to https://container-registry.oracle.com.
2. Sign in to the Oracle Container Registry using an Oracle account.

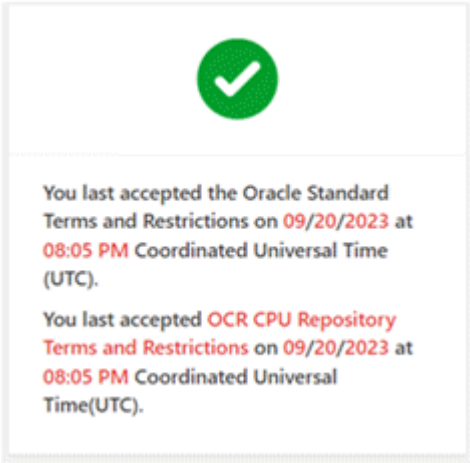
> **Note**
>
> To pull images for licensed software on the Oracle Container Registry, you must have an Oracle account. You can create an Oracle account at https://profile.oracle.com/myprofile/account/create-account.jspx.

3. Click the **Middleware** container, and then click the **fmw-infrastructure_cpu** CPU repository.

   The repository page appears.

4. Accept the Oracle terms and restrictions by clicking **Continue** and, on the next page, clicking **Accept**.

   If successful, you will see something similar to this:



5. On the host system, log in to the Oracle Container Registry using the Podman CLI:

```
podman login container-registry.oracle.com
```

6. When prompted for a user name and password, enter your Oracle credentials.
7. Pull the WebLogic image into your local system using the following command:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
podman pull container-registry.oracle.com/middleware/fmw-infrastructure_cpu:14.1.2.0-jdk21-ol9
```

8. Tag the image with the registry server using the following command, where *RepoHost* and *RepoPort* are the host and port of your private repository:

   ```
   podman tag container-registry.oracle.com/middleware/fmw-infrastructure_cpu:14.1.2.0-jdk21-ol9 RepoHost:RepoPort/middleware/fmw-infrastructure_cpu:14.1.2.0-jdk21-ol9
   ```

9. Push the image to the registry server using the following command, where *RepoHost* and *RepoPort* are the host and port of your private repository:

   ```
   podman push RepoHost:RepoPort/middleware/fmw-infrastructure_cpu:14.1.2.0-jdk21-ol9
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Part III

Configuring and Deploying BRM Cloud Native

This part provides information about configuring and deploying Oracle Communications Billing and Revenue Management (BRM) in your cloud native environment. It contains the following chapters:

- Deploying the BRM Database Schema
- Configuring BRM Server and PDC Services
- Configuring REST Services
- Configuring the BRM Client Services
- Configuring ECE Services
- Deploying and Configuring the Mapper Framework
- Deploying BRM Cloud Native Services
- Deploying into Oracle Cloud Infrastructure
- Uninstalling Your BRM Cloud Native Deployment

Copyright © 2019, 2026, Oracle and/or its affiliates.

6

Deploying the BRM Database Schema

Learn how to deploy a new or existing database schema in the Oracle Communications Billing and Revenue Management (BRM) cloud native environment.

Topics in this document:

- Deploying BRM with a New Database Schema
- Deploying BRM with an Existing Schema

Deploying BRM with a New Database Schema

To deploy BRM with a new BRM and pipeline database schema:

1. If you have not already done so, create and configure your BRM database. See "Creating and Configuring Your BRM Database".
2. Create a new Kubernetes namespace for **oc-cn-init-db-helm-chart**:

   ```
   kubectl create namespace InitDbNameSpace
   ```

3. Create an **override-values.yaml** file for **oc-cn-init-db-helm-chart**, and add keys from Table 6-1.

   > **Note**
   >
   > This documentation uses the **override-values.yaml** file name for ease of use, but you can name the file whatever you want.

4. If you want the init-db Helm chart to create users and tablespaces for you, add the following keys to your file:

   > **Note**
   >
   > Skip this step for production systems or if you pre-created the database user and tablespaces.

   - **db.user**: Specify the user name of the system administrator.
   - **db.password**: Specify the password for the system administrator.
   - **db.port**: Specify the port number of the database server.

   During deployment, the init-db Helm chart uses these values to validate the schema user names and tablespace names that you provide. If any schema users or tablespace names are missing, the init-db Helm chart creates them using the system administrator credentials.

5. If you want to deploy the BRM cloud native schema into a multischema database, do the following in your **override-values.yaml** file:

Copyright © 2019, 2026, Oracle and/or its affiliates.

a. Set the **ocbrm.db.skipPrimary** key to **false**.

b. For each secondary schema in your system, add an **ocbrm.db.multiSchemas.secondary***N* block, where *N* is **1** for the first secondary schema, **2** for the next secondary schema, and so on.

c. In each **ocbrm.db.multiSchemas.secondary***N* block, set the following keys:

> **Note**
>
> If the **host**, **port**, and **service** keys are not defined, the secondary schema uses the same host name, service, and port number as the primary schema.

- **deploy**: Set this to **true** to deploy this secondary schema.
- **host**: Set this to the host name of the secondary schema. This key is optional.
- **port**: Set this to the port number for the secondary schema. This key is optional.
- **service**: Set this to the service name for the secondary schema. This key is optional.
- **schemauser**: Set this to the schema user name.
- **schemapass**: Set this to the schema password.
- **schematablespace**: Set this to the name of the schema tablespace, such as pin01.
- **indextablespace**: Set this to the name of the index tablespace, such as pinx01.

d. Enable account migration between your schemas by setting the **ocbrm.isAmt** key to **true**.

   See "Understanding Account Migration" in *BRM Moving Accounts Between Database Schemas* for more information.

This shows example entries for a BRM database with one primary schema, two secondary schemas, and account migration enabled:

```
ocbrm:
   isAmt: true
ocbrm:
   db:
      host: hostname
      port: 12345
      service: serviceName
      schemauser: pin01
      schemapass: password
      schematablespace: pin01
      indextablespace: pinx01
      skipPrimary: false
      multiSchemas:
         secondary1:
            deploy: true
            schemauser: pin02
            schemapass: password
            schematablespace: pin02
            indextablespace: pinx02
         secondary2:
            deploy: true
            schemauser: pin03
            schemapass: password
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
            schematablespace: pin03
            indextablespace: pinx03
```

6. To commit seed data to the BRM database, modify the init-db container (**configmap_create_obj_2.yaml**) to add the corresponding PCM_OP_CREATE_OBJ input flist. For example:

```
<PCM_OP $PIN_OPNAME=$PIN_CONF_INIT_OPNAME; $PIN_OPFLAGS=0>
0 PIN_FLD_POID           POID [0] $DB_NUMBER /config/recharge_card_type 0 0
0 PIN_FLD_NAME            STR [0] "-"
0 PIN_FLD_PROGRAM_NAME    STR [0] "load_pin_recharge_card_type"
0 PIN_FLD_HOSTNAME        STR [0] "-"
0 PIN_FLD_VERSION         STR [0] "1"
0 PIN_FLD_ACCOUNT_OBJ    POID [0] 0.0.0.1 /account 1 0
</PCM_OP>
```

7. Initialize the BRM database schema by running this command from the **helmcharts** directory:

```
helm install InitDbReleaseName oc-cn-init-db-helm-chart --namespace InitDbNameSpace
--values OverrideValuesFile
```

   where:

   - *InitDbReleaseName* is the release name for **oc-cn-init-db-helm-chart** and is used to track this installation instance.
   - *InitDbNameSpace* is the namespace for **oc-cn-init-db-helm-chart**.
   - *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-init-db-helm-chart**.

   The init-db Helm chart creates an init-db pod and job for each schema.

8. After the init-db Helm chart deploys successfully, delete the Helm chart from your namespace by running this command from the **helmcharts** directory:

```
helm delete InitDbReleaseName -n InitDbNameSpace
```

Table 6-1 lists the keys that directly impact the BRM database schema and pipeline database schema. Add these keys to your **override-values.yaml** file with the same path hierarchy.

> ⚠ **Caution**
>
> Some keys hold sensitive data. They must be handled carefully with controlled access to the file containing its values. Encode all of these values in Base64. See "Secrets" in *Kubernetes Concepts*.

Table 6-1 oc-cn-init-db-helm-chart Keys

Key	Purpose
imageRepository	Specify the registry server where you have pushed images. Typically, in the format “RepoHost:RepoPort/”. The value is added as a prefix to all image names when you install or upgrade Helm charts. This key is empty by default.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 6-1 (Cont.) oc-cn-init-db-helm-chart Keys

Key	Purpose
imagePullSecrets	Specify the name of the secret containing credentials for accessing images from your private image server. This is added to each pod to give it permission to pull the image from your private registry server. See "Creating Secrets for Docker Registry Authorization" for more information.
uniPass	Use this key to apply a uniform password to all BRM cloud native services, including: • Database Schemas • BRM Root Login • Oracle Wallets • WebLogic User To override this password for a specific service, specify a different password in the service's key. **Note**: Use this key for test or demonstration systems only.
db.sslMode	Specify the type of SSL connection required to connect to the database: **TWO_WAY**, **ONE_WAY**, or **NO**. The default is **ONE_WAY**.
db.extDBSSLWalletSecret	Specify the name of the Kubernetes secret containing the SSL database wallet. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
db.user **db.password** **db.port**	Specify the system administrator credentials. Set these keys only if you want the init-db Helm chart to create database users and tablespaces for you. The init-db Helm chart creates any missing schema users and tablespaces using the system administrator credentials. • **db.user**: Specify the user name of the system administrator. • **db.password**: Specify the password of the system administrator. • **db.port**: Specify the port number of the database server. **Note:** If these keys are not included, the init-db Helm chart assumes that the database user and tablespaces have been pre-created and throws an error if it runs into a connection issue.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 6-1 (Cont.) oc-cn-init-db-helm-chart Keys

Key	Purpose
ocbrm.*	Specify the details for deploying BRM: • **imagePullPolicy**: Specify when to pull images: – **IfNotPresent**: Pulls an image only if one is not present locally. This is the default. – **Always**: Always pulls an image. • **isAmt**: Whether account migration is enabled in your BRM multischema database (**true**) or not (**false**). The default is **false**. • **isIPV6Enabled**: Whether IPV6 is enabled in your Kubernetes environment (**true**) or not (**false**). The default is **false**. • **ece_deployed**: Whether to deploy and load ECE configurations into your BRM cloud native environment (**true**) or not (**false**). The default is **true**. • **pdc_deployed**: Whether to deploy and load PDC configurations into your BRM cloud native environment (**true**) or not (**false**). The default is **false**. • **is_upgrade**: Whether to upgrade the database (**true**) or perform a fresh database initialization (**false**). The default is **false**. • **TZ**: Specify the Linux time zone. The default is **UTC**. • **LOG_LEVEL**: Specify the log level. The default is **1**. • **isSSLEnabled**: Whether TLS is enabled for BRM components. The default is **true**. • **sslMode**: Specify the TLS authentication mode for BRM component connections: **ONE_WAY** or **TWO_WAY**. The default is **ONE_WAY**. • **cmSSLTermination**: Whether the CM terminates TLS and disables TLS between the CM and DMs/EMs – **true**: The CM is the SSL endpoint. Set it to **true** when using a custom TLS certificate for the CM. When set to **true**, also set the **isSSLEnabled** key to **true**. – **false**: SSL is enabled across all components, from client to CM to DMs and EMs. This is the default. • **extCustomSSLWalletSecret**: Specify the name of an external Kubernetes secret containing customer-provided TLS wallet material. BRM uses this secret as a shared wallet source when a component-specific wallet secret is not configured. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **isHPAEnabled**: Whether to enable autoscaling (**true**) or not (**false**). The default is **false**. • **refreshInterval**: Specify the interval for refreshing the PCM context to accept a request on a new CM. • **terminationGracePeriodSeconds**: Specify how long to wait before terminating the pod. Increase the value for long-running transactions. • **pcpReconnectDelayOnSocketError**: Specify how long to wait for a socket reconnection. • **pcpConnectRetryDelayOnError**: Specify the amount of time for retrying a connection after the PCP connection fails. • **EnableSecurityContext**: Whether to enable the Security Context in the cluster. The default is **false**. • **root_key_rotate**: Whether to rotate the BRM root key. The default is **false**. For more information, see "Rotating the BRM Root Key" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 6-1 (Cont.) oc-cn-init-db-helm-chart Keys

Key	Purpose
	• **brm_root_pass**: Specify the BRM root password. The value can be per user. You must generate a secret. Add this key to your **override-values.yaml** file with the same path hierarchy. • **rotate_brm_role_passwords**: Whether to rotate the BRM role passwords. The default is **false**.
ocbrm.brm_role_pass.*	Specify the initial passwords for each BRM role. The roles grant users the permission to access different BRM components, such as Billing Care or Pipeline Configuration Center (PCC). **Note:** You must set all of these passwords when the **unipass** key is not set. You set passwords for the following roles: • **acct_recv.0.0.0.1**: Accounts receivable • **bc_client.0.0.0.1**: Billing Care • **bill_inv_pymt_sub.0.0.0.1**: Invoice payments • **billing.0.0.0.1**: Billing • **boc_client.0.0.0.1**: Business Operations Center • **collections.0.0.0.1**: Collections • **crypt_utils.0.0.0.1**: Encryption utilities • **cust_mgnt.0.0.0.1**: Customer management • **invoicing.0.0.0.1**: Invoicing • **java_client.0.0.0.1**: Java clients • **load_utils.0.0.0.1**: Load utilities • **payments.0.0.0.1**: Payments • **pcc_client.0.0.0.1**: PCC • **rerating.0.0.0.1**: Rerating • **rsm.0.0.0.1**: BRM REST Services Manager • **super_user.0.0.0.1**: Super User • **ui_client.0.0.0.1**: All thick clients • **ece.0.0.0.1**: ECE **Note**: After you deploy or upgrade the database schema, you should rotate the role passwords regularly to improve security. For more information, see "Rotating BRM Role Passwords" in *BRM Cloud Native System Administrator's Guide*.
ocbrm.wallet.config	The BRM configuration wallet used to securely store sensitive BRM configuration information, such as passwords, database credentials, and connection information. The wallet is created at runtime and is available at **/oms/wallet/config**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 6-1 (Cont.) oc-cn-init-db-helm-chart Keys

Key	Purpose
ocbrm.db.*	Specify the details for connecting to a shared database. The keys in this section take precedence over other database connection keys. Add these keys only if your system uses a shared database: • **host**: Specify the host name of the machine on which the BRM database is configured. • **port**: Specify the port on which the BRM database is configured. • **service**: Specify the BRM database service name. • **sslMode**: Specify the type of SSL connection required to connect to the database: **TWO_WAY**, **ONE_WAY**, or **NO**. • **extDBSSLWalletSecret**: Specify the name of the secret containing the SSL database wallet. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **walletPassword**: Specify the password for accessing the database certificates from the TrustStore and KeyStore. • **walletType**: Specify the type of file specified as the TrustStore or KeyStore for database SSL connections: **SSO** or **pkcs12**. • **enable_partition**: Whether partitioning is enabled in your BRM database: **Yes** or **No**. The default is **Yes**. • **enable_delayed_partition**: Whether delayed partitioning is enabled for all subclasses of the **/event** storable class: **Yes** or **No**. The default is **Yes**. • **storage_model**: Specify the size of the BRM database tablespaces: **Test** (less than 700 MB), **Small** (less than 1.5 GB), **Medium** (less than 30 GB), or **Large** (greater than 30 GB). • **schemauser**: Specify the user name for the primary BRM schema. • **schemapass**: Specify the password for the database schema. • **schematablespace**: Specify the name of the schema tablespace. • **indextablespace**: Specify the name of the index tablespace. • **nls_lang**: The language, territory, and character set. Set this to **American_America.***characterset*, where *characterset* is either **UTF8** or **AL32UTF8**. **Note:** You must use **American_America** as the language and territory, regardless of your locale. • **pipelineschemauser**: Specify the BRM pipeline schema user name. • **pipelineschemapass**: Specify the BRM pipeline schema password. You must generate a secret. Add this key to your **override-values.yaml** file with the same path hierarchy. • **pipelineschematablespace**: Specify the name of the tablespace for the BRM pipeline schema. This field is case-sensitive. • **pipelineindextablespace**: Specify the name of the index tablespace for the BRM pipeline schema. This field is case-sensitive. • **skipPrimary**: Whether to skip the deployment of the primary schema: **False** (deploy the primary schema) or **True** (do not deploy the primary schema). Set **skipPrimary** to **true** only if you are adding a schema to an existing BRM multischema system. See "Adding Schemas to a Multischema System" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 6-1 (Cont.) oc-cn-init-db-helm-chart Keys

Key	Purpose

ocbrm.db.multiSchemas.secondary*N*	Specify the details for connecting to your secondary schemas, where *N* is **1** for the first secondary schema, **2** for the next secondary schema, and so on.
	Add this block only if your BRM database contains multiple schemas. This section is commented out by default:
	`multiSchemas:`
	`  secondary1:`
	`  deploy: true`
	`  host: localhost`
	`  port: 1521`
	`  service: pindb`
	`  schemauser:`
	`  schemapass:`
	`  schematablespace:`
	`  indextablespace:`

Deploying BRM with an Existing Schema

You can deploy BRM cloud native with an existing BRM database schema by using either the default Oracle images or custom BRM server images.

Starting in BRM 16.26.0, BRM stores the root-key wallet in the BRM database and retrieves it from the database when required.

If you enable TLS, provide the required TLS identity and trust material by using the supported Kubernetes secret configuration.

To deploy BRM with an existing schema:

1. Complete the database upgrade before you deploy the BRM services.

   You do not need to copy the root-key wallet into the Helm chart after the upgrade.

2. If you use custom BRM server images, create an image for each BRM server component in the installed BRM staging area that you used to initialize the database.

   Do not include populated wallets, private keys, sample CA certificates, or default credentials in your custom images or image-build context.

   If you use the default Oracle images, skip this step.

3. In your **override-values.yaml** file for **oc-cn-helm-chart**, configure the connection information and credentials for the existing BRM database schema. Also configure the following keys as applicable:

   - **ocbrm.use_oracle_brm_images**: If you use custom BRM Server images, set this to **false**. If you use the default Oracle images, set this to **true**.
   - **ocbrm.db.queuename**: Set this to the queue name in the existing BRM database.
   - **ocbrm.db.roamqueuename**: If you use a database queue for Roaming Pipeline, create the queue by following the instructions in "Creating Additional Queues for Multischema BRM Systems" in *BRM Installation Guide*. Set the **roamqueuename** key to the name of the queue that you create.

Copyright © 2019, 2026, Oracle and/or its affiliates.

4. If you enable TLS and use external Kubernetes secrets, pre-create the secrets that contain the required TLS wallet material.

   Configure a component-specific secret by using the appropriate **ocbrm.***component***.extUserSSLWalletSecret** key, or configure a shared wallet secret by using the **ocbrm.extCustomSSLWalletSecret** key, as appropriate for your deployment. BRM uses the configured external wallet secret instead of obtaining TLS wallet material from the container image.

   You do not need to create a Kubernetes secret for the root-key wallet because BRM stores the root-key wallet in the database.
   For more information about configuring wallet secrets, see "Managing Wallet and KeyStore Certificates".

5. Deploy **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

7

Configuring BRM Server and PDC Services

Learn how to configure the Oracle Communications Billing and Revenue Management (BRM) server and Pricing Design Center (PDC) in your cloud native environment.

Topics in this document:

- About Configuring BRM Cloud Native Services
- Creating Secrets for Docker Registry Authorization
- Managing Wallet and KeyStore Certificates
- Configuring Global Values
- Specifying the BRM Services to Deploy
- Configuring the BRM Server
- Configuring BRM for a Multischema Database
- Configuring Pricing Design Center
- Configuring Pipeline Configuration Center

About Configuring BRM Cloud Native Services

You configure and deploy BRM cloud native services, such as BRM, PDC, and PCC, by using the BRM Helm chart (**oc-cn-helm-chart**). YAML descriptors in the **oc-cn-helm-chart/templates** directory use the **oc-cn-helm-chart/values.yaml** file for most of the values. The **values.yaml** file itself includes comments that describe each key. You can override the values by creating an **override-values.yaml** file.

Oracle recommends that you read the **values.yaml** file at least once to become familiar with all of the options available.

Creating Secrets for Docker Registry Authorization

You can automatically pull images from your private container registry by creating an **ImagePullSecrets**, which contains a list of authorization tokens (or Secrets) for accessing a private container registry. You then add references to the **ImagePullSecrets** in your BRM Helm chart's **override-values.yaml** file. This allows pods to submit the Secret to the private container registry whenever they want to pull images.

Automatically pulling images from a private container registry involves these high-level steps:

1. Create a Secret outside of the Helm chart by entering this command:

   ```
   kubectl create secret docker-registry SecretName --docker-server=RegistryServer --
   docker-username=UserName --docker-password=Password --namespace NameSpace
   ```

   where:

   - *SecretName* is the name of your Kubernetes Secret
   - *RegistryServer* is your private container registry's FQDN (*repoHost*:*repoPort*)

Copyright © 2019, 2026, Oracle and/or its affiliates.

- *UserName* and *Password* are your private container registry's user name and password
- *NameSpace* is the namespace you will use for installing BRM Helm charts

For example:

```
kubectl create secret docker-registry cgbu-docker-registry --docker-
server=mydockerimages.com:2660/ --docker-username=xyz --docker-password=password --
namespace oms
```

2. Add the **imagePullSecrets** key to your **override-values.yaml** file for **oc-cn-helm-chart**:

```
imagePullSecrets:
    -name "SecretName1"
    -name "SecretName2"
```

3. Add the **imageRepository** key to your **override-values.yaml** file:

```
imageRepository: "RegistryServer"
```

4. Deploy **oc-cn-helm-chart**.

Managing Wallet and KeyStore Certificates

BRM cloud native does not include populated TLS wallets, private keys, or sample certificate authority (CA) certificates in its container images. When you enable TLS, you must provide certificates and trust material approved by your organization and manage them throughout their lifecycle, including renewal and rotation.

BRM cloud native uses separate wallets for configuration data, TLS trust, and TLS identity, as shown in Table 7-1.

Table 7-1 BRM Cloud Native Wallets

Wallet	Location	Description
Configuration wallet	**/oms/wallet/config**	Stores BRM keys, database credentials, Infranet connection strings, and related configuration. The BRM entrypoint script creates this wallet at runtime.
Client wallet	**/oms/wallet/client**	Stores trusted certificates that BRM components use to validate certificates presented by remote TLS endpoints.
Server wallet	**/oms/wallet/server**	Stores the certificate and private key that a BRM component uses to establish its TLS identity.

You can provision TLS wallets and keystores in either of the following ways:

- Pre-create the required wallets and keystores as Kubernetes secrets before you deploy BRM cloud native. Configure the appropriate external secret keys in your **override-values.yaml** file. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide* for more information.
- Provide the required wallet and keystore files in the appropriate locations in the BRM cloud native Helm charts. During installation, the Helm charts create Kubernetes secrets from the files that you provide.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Selecting the TLS Wallet Source

When you enable TLS, BRM selects the TLS wallet for each component from the following sources, in order of precedence:

1. **Component-specific wallet secret**: BRM first uses the Kubernetes secret specified by **ocbrm.***component***.extUserSSLWalletSecret**, where *component* identifies the BRM component. For example, set ocbrm.dm_oracle.extUserSSLWalletSecret to specify a TLS wallet for the Oracle DM.
2. **Shared external wallet secret**: If you do not configure a component-specific wallet secret, BRM uses the Kubernetes secret specified by **ocbrm.extCustomSSLWalletSecret**.
3. **Chart-created wallet secret**: If you do not configure either external secret, the Helm chart creates the **custom-ssl-wallet** Kubernetes secret from the **brm_custom_wallet/cwallet.sso** file and BRM uses that secret.

A component-specific wallet secret takes precedence over the shared wallet sources. You can therefore configure a separate TLS wallet for an individual BRM component while allowing other components to use a shared wallet.

If you enable TLS but do not provide any of these wallet sources, the Helm chart installation fails.

Configuring BRM Client Applications for Two-Way TLS

When you configure BRM to use two-way TLS, BRM client applications that connect to the CM over PCP must present their own certificate to CM for authentication.

To configure a BRM client application for two-way TLS:

1. Provide a TLS client wallet that contains:
   - The trusted certificate authority (CA) certificates required to validate the CM certificate.
   - A user certificate and corresponding private key that CM can use to authenticate the client application.
2. Enable client authentication in the application's **Infranet.properties** configuration by setting:

   ```
   infranet.pcp.ssl.client_auth=true
   ```

   In the application's **override-values.yaml** file, add **infranet.pcp.ssl.client_auth=true** to the **infranet.addOnProperties** setting. This setting adds the property to the application's **Infranet.properties** configuration.

For one-way TLS, CM presents its certificate to the BRM client application, and the client validates the certificate using the trusted CA certificates in its wallet. The client does not present a certificate to CM. Therefore, do not add **infranet.pcp.ssl.client_auth=true**, and a client certificate and private key are not required in the client wallet.

Configuring Global Values

Table 7-2 lists the keys that apply to all BRM components. To set or change the values, add them to your **override-values.yaml** file for **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-2 Global Keys in Values.yaml File

Key	Description
imageRepository	The registry server where you have pushed images. Typically, in the format “RepoHost:RepoPort/”. The value is added as a prefix to all image names when you install or upgrade Helm charts. This key is empty by default.
imagePullSecrets	The name of the Secret that contains credentials for accessing images from your private image server. This is added to each pod to give it permission to pull the image from your private registry server. See "Creating Secrets for Docker Registry Authorization" for more information. This key is empty by default.
uniPass	Use this key to apply a uniform password to all BRM cloud native services, including: • Database Schemas • BRM Root Login • BRM Role Passwords • Oracle Wallets • WebLogic User To override this password for a specific service, specify a different password in the service's key. **Note**: Use this key for test or demonstration systems only.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-2 (Cont.) Global Keys in Values.yaml File

Key	Description
db.*	The details for connecting to a shared database. The keys in this section take precedence over other database connection keys. Add these keys only if your system uses a shared database: • **sslMode**: The type of SSL connection required for connecting to the database: – **TWO_WAY**: Two-way SSL authentication is required. In this case, both the client and server must authenticate each other's identity. – **ONE_WAY**: One-way SSL authentication is required. In this case, the client must authenticate the server's identity. This is the default. – **NO**: SSL authentication is not required. • **extDBSSLWalletSecret**: The name of the Secret containing the SSL database wallet. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **host**: The host name or IP address of the database server. • **port**: The port number of the database server. • **user**: The user name of the database administrator. • **password**: The password of the database system administrator. • **serviceName**: The service name that identifies the database. • **role**: The role assigned to the DBA user. • **walletPassword**: The password for accessing the certificates from the TrustStore and KeyStore. This is required if **sslMode** is set to **ONE_WAY** or **TWO_WAY**. • **walletType**: The type of file specified as the TrustStore or KeyStore for SSL connections: **SSO** or **pkcs12**. • **sslServerCertDN**: The Distinguished Name (DN) of the TLS certificate presented by the Oracle Database server. The value must match the DN in the database server certificate. Set this key when database TLS authentication is enabled.
security.*	The details for setting security in BRM cloud native: • **tlsVersions**: The list of TLS versions supported. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**. • **java.overrideSecurityProperties**: Whether to override the default Java security property (**true**) or not (**false**). The default is **true**.
monitoring.prometheus.jmx_exporter.enable	Whether to enable the JMX exporter for Prometheus (**true**) or not (**false**). The default is **false**. See "Monitoring BRM Cloud Native Services" in *BRM Cloud Native System Administrator's Guide* for more information.
monitoring.prometheus.operator.*	The details for monitoring BRM cloud native services using Prometheus: • **enable**: Whether to use Prometheus Operator (**true**) or standalone Prometheus (**false**). The default is **false**. • **namespace**: The namespace in which Prometheus Operator is deployed. The default is **prometheus**. • **release**: The release name for Prometheus Operator. The default is **prometheus**. See "Monitoring BRM Cloud Native Services" in *BRM Cloud Native System Administrator's Guide* for more information.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Specifying the BRM Services to Deploy

Some BRM cloud native services are enabled by default, while others are disabled. Ensure that your **override-values.yaml** file is set up to deploy the services that you want to include in your BRM cloud native environment.

BRM Cloud Native Services Enabled by Default

Table 7-3 lists the BRM cloud native services that are deployed by default. To exclude them from your deployment, set the keys to **false** in your **override-values.yaml** file for **oc-cn-helm-chart**.

Table 7-3 BRM Services Enabled By Default

BRM Service	override-values.yaml Key
Batch Pipeline	**ocbrm.batchpipe.isEnabled**
Billing Care	**ocbc.bc.isEnabled**
Billing Care REST API	**ocbc.bcws.isEnabled**
Business Operations Center	**ocboc.boc.isEnabled**
BRM REST Services Manager	**ocrsm.rsm.isEnabled**
Collections Configuration Center	**occcc.ccc.isEnabled**
Connection Manager	**ocbrm.cm.isEnabled**
Oracle Data Manager	**ocbrm.dm_oracle.isEnabled**
Pipeline Configuration Center	**ocpcc.pcc.isEnabled**
Pricing Design Center	**ocpdc.isEnabled**
Rated Event (RE) Loader Daemon	**ocbrm.rel_daemon.isEnabled**
Realtime Pipeline	**ocbrm.realtimepipe.isEnabled**

BRM Cloud Native Services Disabled By Default

Table 7-4 lists the BRM cloud native services that are *not* deployed by default. To include them in your BRM cloud native deployment, set the keys to **true** in your **override-values.yaml** file for **oc-cn-helm-chart**.

Table 7-4 BRM Services Disabled By Default

BRM Service	override-values.yaml Key
Batch Controller	**ocbrm.batch_controller.isEnabled**
Billing Care REST API SDK	**ocbc.bcws.sdk.isEnabled**
Billing Care SDK	**ocbc.bc.sdk.isEnabled**
BRM Apps Jobs	**ocbrm.brm_apps.job.isEnabled**
BRM SDK	**ocbrm.brm_sdk.isEnabled**
Email Data Manager	**ocbrm.dm_email.isEnabled**
Enterprise Application Integration (EAI) Data Manager	**ocbrm.dm_eai.isEnabled**
Event Stream Processor	**ocbrm.event_stream_processor.isEnabled**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-4 (Cont.) BRM Services Disabled By Default

BRM Service	override-values.yaml Key
Invoicing Formatter	**ocbrm.formatter.isEnabled**
Invoicing Data Manager	**ocbrm.dm_invoice.isEnabled**
Kafka Data Manager	**ocbrm.dm_kafka.isEnabled**
LDAP Data Manager	**ocbrm.dm_ldap.isEnabled**
Paymentech Data Manager	**ocbrm.dm_fusa.isEnabled**
PDC REST Services Manager	**ocpdcrsm.isEnabled**
Provisioning Data Manager	**ocbrm.dm_prov_telco.isEnabled**
RE Loader Daemon Job	**ocbrm.rel_daemon.job.isEnabled**
RE Loader Manager Job	**ocbrm.rel_manager.job.isEnabled**
Rated Event Manager	**ocbrm.rem.isEnabled**
Roaming Pipeline	**ocbrm.roampipe.isEnabled**
Taxation Gateway	**ocbrm.taxation_gateway.isEnabled**
Vertex Data Manager	**ocbrm.dm_vertex.isEnabled**
Webhook	**webhook.isEnabled**
Web Services Manager	**ocbrm.wsm.soap.isEnabled**

Configuring the BRM Server

To configure the BRM server to run in your cloud native environment, you override the BRM server-specific keys in the **values.yaml** file for **oc-cn-helm-chart**. Table 7-5 lists the keys that directly impact BRM Server pods. Add these keys to your **override-values.yaml** file with the same path hierarchy.

> **Note**
>
> You can optionally deploy a simple demonstration version of BRM cloud native by using the sample **override_values.yaml** file that is packaged with **oc-cn-helm-chart**. This sample override file contains the bare minimum keys that you need to update to create a simple BRM cloud native system with the following services enabled by default: Account Synchronization DM, Batch Pipeline, CM, Oracle DM, RE Loader, Realtime Pipeline, Billing Care, Billing Care REST API, Business Operations Center, and PDC.

Table 7-5 BRM Server Keys

Key	Path in values.yaml File	Description
imagePullPolicy	**ocbrm**	The image pull policy for BRM server images. The default is IfNotPresent.
isAmt	**ocbrm**	Whether account migration is enabled in your BRM database (**true**) or not (**false**). The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
isIPV6Enabled	**ocbrm**	Whether IPV6 is enabled in your Kubernetes environment (**true**) or not (**false**). The default is **false**.
ece_deployed	**ocbrm**	Whether ECE is going to be deployed in your BRM cloud native environment (**true**) or not (**false**). The default is **true**.
pdc_deployed	**ocbrm**	Whether PDC is going to be deployed: • **true**: Configuration data is *not* loaded into the BRM database. Only mandatory configuration records are loaded into the BRM database for starting the realtime pipeline pod. For the batch pipeline and roaming pipeline pods, you must load the required configuration data using PDC before deploying the pods. • **false**: Configuration data is loaded into the BRM database during deployment. This is the default.
use_oracle_brm_images	**ocbrm**	Whether to use the default BRM images (**true**) or not (**false**). Set this to **false** if you are building custom images. The default is **true**.
is_upgrade	**ocbrm**	Whether to upgrade the BRM database (**true**) or perform a fresh database initialization or rollback (**false**). The default is **false**. See "Upgrading Your BRM Cloud Native Services".
is_rollback	**ocbrm**	Whether to roll back the BRM database upgrade (**true**) or not (**false**). The default is **false**.
virtual_time.*	**ocbrm**	The details for configuring the **pin_virtual_time** utility: • **enabled**: Whether to enable **pin_virtual_time**. The default is **false**. • **sync_pvt_time**: Set this to the number of seconds between each synchronization of **pin_virtual_time** with all pods. The default is **60** seconds. • **volume.storage**: The storage size of the volume. • **volume.createOption**: By default, the virtual-time pod uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.
TZ	**ocbrm**	The time zone for BRM server pods. The default is UTC.
LOG_LEVEL	**ocbrm**	The logging level for BRM server components. The default is 1.
isSSLEnabled	**ocbrm**	Whether SSL is enabled for BRM (**true**) or disabled (**false**). Set this key to **true** when you use custom TLS certificates. The default is **true**.
sslMode	**ocbrm**	The SSL mode for BRM connections: **ONE_WAY** or **TWO_WAY**. The default is **ONE_WAY**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
cmSSLTermination	**ocbrm**	Whether to make the CM the SSL endpoint for the BRM cloud native deployment: • **true**: The CM is the SSL endpoint. In this case, TLS can be enabled only between BRM client applications and the CM. TLS is disabled between CM and all downstream components such as DMs and EMs. Communication between external clients and the CM will still be encrypted by TLS 1.2. This setting can increase performance, because it eliminates the overhead needed to handle TLS before processing the PCP packets. When set to **true**, also set the **isSSLEnabled** key to **true**. • **false**: SSL is enabled across all components, from client to CM to DMs and EMs. This is the default.
extCustomSSLWalletSecret	**ocbrm**	The name of the external Kubernetes Secret for the custom SSL wallet. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
isHPAEnabled	**ocbrm**	Whether horizontal pod autoscaling is enabled. The default is false.
refreshInterval	**ocbrm**	The interval for refreshing PCM contexts so they can accept requests on a new CM.
terminationGracePeriodSeconds	**ocbrm**	The time, in seconds, that Kubernetes waits before terminating a pod. Increase this value for long-running transactions.
pcpReconnectDelayOnSocketError	**ocbrm**	The delay before retrying a socket connection after a socket error.
pcpConnectRetryDelayOnError	**ocbrm**	The delay before retrying a PCP connection after a connection failure.
root_key_rotate	**ocbrm**	Whether to rotate the BRM root key (**true**) or not (**false**). The default is **false**. See "Rotating the BRM Root Key" in *BRM Cloud Native System Administrator's Guide*.
brm_root_pass	**ocbrm**	The root password. See "Rotating the BRM Root Password" in *BRM Cloud Native System Administrator's Guide*.
rotate_password	**ocbrm**	Whether to rotate the BRM root password: • **true**: The BRM root password is replaced with the one specified in the **new_brm_root_password** key. • **false**: The BRM root password is not changed. This is the default. See "Rotating the BRM Root Password" in *BRM Cloud Native System Administrator's Guide*.
new_brm_root_password	**ocbrm**	The new BRM root password. Use this key only when **ocbrm.rotate_password** is set to **true**. See "Rotating the BRM Root Password" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
rotate_brm_role_passwords	**ocbrm**	Whether to rotate the BRM role passwords. The default is **false**. See "Rotating BRM Role Passwords" in *BRM Cloud Native System Administrator's Guide*.
brm_role_pass.*	**ocbrm**	The passwords for each BRM role. The roles grant users the permission to access different BRM components, such as Billing Care or Pipeline Configuration Center (PCC). **Note:** You must set all of these passwords when the **unipass** key is not set. You set passwords for the following roles: • **acct_recv.0.0.0.1**: Accounts receivable • **bc_client.0.0.0.1**: Billing Care • **bill_inv_pymt_sub.0.0.0.1**: Invoice payments • **billing.0.0.0.1**: Billing • **boc_client.0.0.0.1**: Business Operations Center • **collections.0.0.0.1**: Collections • **crypt_utils.0.0.0.1**: Encryption utilities • **cust_center.0.0.0.1**: Customer Center • **cust_mgnt.0.0.0.1**: Customer management • **invoicing.0.0.0.1**: Invoicing • **java_client.0.0.0.1**: Java clients • **load_utils.0.0.0.1**: Load utilities • **payments.0.0.0.1**: Payments • **pcc_client.0.0.0.1**: PCC • **rerating.0.0.0.1**: Rerating • **rsm.0.0.0.1**: BRM REST Services Manager • **super_user.0.0.0.1**: Super User • **ui_client.0.0.0.1**: All UI clients • **ece.0.0.0.1**: ECE The passwords in this key must match the passwords in **oc-cn-init-db-helm-chart**. See "Rotating BRM Role Passwords" in *BRM Cloud Native System Administrator's Guide*.
extOmsSchemaPasswordSecret	**ocbrm.secret**	The name of the Kubernetes Secret containing the Secret for the BRM database schema. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
wallet.config	**ocbrm**	The BRM configuration wallet used to securely store sensitive BRM configuration information, such as passwords, database credentials, and connection information. The wallet is created at runtime and is available at **/oms/wallet/config**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
cm.*	**ocbrm**	The details for deploying the cm pod: • **isEnabled**: Whether to enable the CM. The default is **true**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.replicaCount**: The number of replicas to create of the cm pod. The default is **1**. • **deployment.enable_publish**: Whether to publish events (**1**) or not (**0**). The default is **0**. • **deployment.enable_prefs_enrichment**: Whether to enrich notifications with subscriber preferences (**true**) or not (**false**). The default is **false**. • **deployment.prefs_enabled_publisher_list**: The list of publishers with enrichment enabled. The default is **0.0.9.6**. • **deployment.prefs_phone_no_location**: Where to retrieve the phone numbers for subscribers. The default is **0**. • **deployment.provisioning_enabled**: Whether to enable provisioning of service orders (**true**) or not (**false**). The default is **false**. • **deployment.simulate_agent**: Whether to publish service orders (**0**) or not (**1**). The default is **1**. • **deployment.perflib_enabled**: Whether to enable monitoring of the cm service using the performance library (Perlib). The default is **true**. See "Monitoring BRM Cloud Native Services" in *BRM Cloud Native System Administrator's Guide*. • **service.type**: The service type. The default is **ClusterIP**. • **custom_files.enable**: Whether to expose the **oc-cn-helm-chart/cm_custom_files** directory as a ConfigMap (**true**) or not (**false**). The default is **false**. See "Exposing Directories as ConfigMaps" in *BRM Cloud Native System Administrator's Guide*. • **custom_files.extCustomFilesCM**: The name of the ConfigMap for the external CM custom files. • **resources.***: The minimum and maximum CPU and memory resources for the cm pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **hpaValues.***: The details for scaling up or down the number of pod replicas in your deployment based on the pod's CPU or memory utilization. By default, the Horizontal Pod Autoscaler is disabled. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*. • **deployment.imageName**: The name of the CM container image. The default is **cm**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **deployment.imageTag**: The tag for the CM container image. • **nodeSelector**: The rules for deploying the CM pod on specific nodes. • **affinity**: The rules for running the CM pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.
volume.storage	**ocbrm.custom_job_files**	The storage size of the volume. The default is **50Mi**.
eai_js	**ocbrm**	The details for deploying the EAI Java Server. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the EAI Java Server pod. • **extPayloadCM**: The name of the ConfigMap for the external payload files. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
dm_oracle.*	**ocbrm**	The details for deploying the dm-oracle pod: • **isEnabled**: Whether to enable the Oracle DM. The default is **true**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.replicaCount**: The number of replicas to create of the dm_oracle pod. The default is **1**. • **deployment.perflib_enabled**: Whether to enable monitoring of the dm_oracle service using Perlib. The default is **true**. See "Monitoring BRM Cloud Native Services" in *BRM Cloud Native System Administrator's Guide*. • **config.totalFrontEnds**: The total number of front end processes. The default is **4**. • **config.totalBackEnds**: The total number of back end processes. The default is **6**. • **config.connectionsPerFrontEnd**: The number of connections for each front end process. The default is **16**. • **config.totalTransBackEnds**: The total number of back end transactions. The default is **4**. • **config.dmSequenceCacheSize**: The number of POIDs to cache when each instance of an Oracle DM is started. The default is **10000**. • **config.maxStatementCache**: The maximum size of the statement cache. The default is **256**. • **config.sharedMemoryBigSize**: The size of the DM shared memory for "big" shared memory structures, such as those used for large searches. The default is **268435456**. • **config.sharedMemorySegmentSize**: The size of the DM shared memory segment. The default is **2147483648**. • **secondaryConfig.***: The configuration for the secondary Oracle DM. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **hpaValues.***: The details for scaling up or down the number of pod replicas in your deployment based on the pod's CPU or memory utilization. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*. • **deployment.imageName**: The name of the Oracle DM container image. The default is **dm_oracle**. • **deployment.imageTag**: The tag for the Oracle DM container image.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **deployment.DM_DEBUG**: The Oracle DM debug setting. The default is **0x0**. • **deployment.DM_DEBUG2**: The Oracle DM secondary debug setting. The default is **0x0**. • **deployment.DM_DEBUG3**: The Oracle DM third debug setting. The default is **0x0**. • **deployment.DM_DEBUG5**: The Oracle DM fifth debug setting. The default is **0x0**. • **nodeSelector**: The rules for deploying the Oracle DM pod on specific nodes. • **affinity**: The rules for running the Oracle DM pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
dm_kafka.*	**ocbrm**	The details for configuring the Kafka DM: • **isEnabled**: Whether to enable the Kafka DM (**true**) or not (**false**). The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **kafkaAsyncMode**: Whether to use asynchronous mode, which logs business events that failed to publish (**true**) or synchronous mode, which returns errors for business events that failed to publish (**false**). The default is **false**. • **maxBlock**: The maximum block size. The default is **3000**. • **extKafkaKeystoreSecret**: The name of the Kubernetes Secret containing the KeyStore certificate files for Kafka DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **deployment.imageName**: The name of the dm_kafka image. The default is **dm_kafka**. • **deployment.imageTag**: The tag name for the dm_kafka image. • **deployment.replicaCount**: The number of replicas to create of the dm_kafka pod. The default is **1**. • **deployment.jvmOpts**: The Java options to configure, such as heap memory and JVM configurations. • **deployment.kafka_bootstrap_server_list**: Set this to a comma-separated list of addresses for the Kafka brokers in this format: *hostname1*:*port1*, *hostname2*:*port2*. • **deployment.poolSize**: Set this to the number of threads that can run in the JS server to accept requests from the CM. Enter a number from 1 through 2000. The default is **64**. • **deployment.topicName**: Set this to the name of the default Kafka topic. The default name is **BRMTopic**. • **deployment.topicFormat**: Set this to the format of the payload that is published to the default Kafka topic: **XML** or **JSON**. The default is **XML**. • **deployment.topicStyle**: The style of XML payloads: **ShortName**, **CamelCase**, **NewShortName**, or **OC3CNotification**. The default is **CamelCase**. • **deployment.isSecurityEnabled**: Whether to enable SSL between Kafka DM and Kafka Server (**true**) or not (**false**). The default is **false**. • **deployment.trustStorePassword**: The TrustStore password in Base64 format. Required only if SSL is enabled. • **deployment.keyStorePassword**: The KeyStore password in Base64 format. Required only if SSL is enabled.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **deployment.keyPassword**: The password of a key in the KeyStore in Base64 format. Required only if SSL is enabled. • **deployment.password**: The password in Base64 format. Required only if SSL is enabled. • **volume.storage**: The storage size of the volume. • **volume.createOption**: By default, the Kafka DM uses dynamic volumes. To use a static volume instead, add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **hpaValues.***: The details for scaling up or down the number of pod replicas in your deployment based on the pod's CPU or memory utilization. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*. • **secret.extDmKafkaPasswordSecret**: The name of the external Kubernetes Secret containing the passwords for the Kafka DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **secret.extDmKafkaKeystoreSecret**: The name of the external Kubernetes Secret containing the Keystore certificate for the Kafka DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **deployment.isClientKeystoreEnabled**: Whether to enable the client keystore for Kafka DM. The default is **false**. • **deployment.securityProtocol**: The security protocol for connecting Kafka DM to Kafka. • **deployment.saslMechanism**: The SASL authentication mechanism for Kafka DM. • **deployment.saslLoginModule**: The SASL login module for Kafka DM. • **deployment.saslUsername**: The SASL user name for Kafka DM. • **deployment.clientId**: The client ID used for Kafka authentication. • **deployment.clientSecret**: The client secret used for Kafka authentication. • **nodeSelector**: The rules for deploying the Kafka DM pod on specific nodes. • **affinity**: The rules for running the Kafka DM pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		For more information about integrating BRM cloud native with a Kafka Server, see "Integrating with Kafka Servers" in *BRM Cloud Native System Administrator's Guide*.
dm_email.*	**ocbrm**	The details for configuring the Email DM: • **isEnabled**: Whether to enable the email DM. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the dm-email pod. • **config.totalFrontEnds**: The total number of front end processes. The default is **4**. • **config.totalBackEnds**: The total number of back end processes. The default is **8**. • **config.connectionsPerFrontEnd**: The number of connections for each front end process. The default is **16**. • **config.sharedMemoryBigSize**: The size of the DM shared memory for "big" shared memory structures, such as those used for large searches. The default is **512k**. • **config.sharedMemorySize**: The amount of shared memory allocated for use by the email DM processes. The default is **4m**. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
dm_invoice.*	**ocbrm**	The details for configuring the Invoice DM: • **isEnabled**: Whether to enable the Invoice DM. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the dm-invoice pod. • **config.totalFrontEnds**: The total number of front end processes. The default is **2**. • **config.totalBackEnds**: The total number of back end processes. The default is **6**. • **config.connectionsPerFrontEnd**: The number of connections for each front end process. The default is **16**. • **config.totalTransBackEnds**: The total number of back end transactions. The default is **4**. • **config.dmSequenceCacheSize**: The number of POIDs to cache when each instance of an Invoice DM is started. The default is **10000**. • **config.maxStatementCache**: The maximum size of the statement cache. The default is **256**. • **config.sharedMemoryBigSize**: The size of the DM shared memory for "big" shared memory structures, such as those used for large searches. The default is **268435456**. • **config.sharedMemorySegmentSize**: The size of the DM shared memory segment. The default is **2147483648**. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the Invoice DM pod on specific nodes. • **affinity**: The rules for running the Invoice DM pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
dm_ldap.*	**ocbrm**	The details for configuring LDAP DM: • **isEnabled**: Whether to enable the LDAP DM. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the LDAP DM. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the **dm_ldap** pod on specific nodes. • **affinity**: The rules for running the **dm_ldap** pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **secret.extDmLdapPasswordSecret**: The name of the Kubernetes Secret containing the passwords for the LDAP DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
dm_prov_telco.*	**ocbrm**	The details for configuring Provisioning DM: • **isEnabled**: Whether to enable the Provisioning DM. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the Provisioning DM. • **volume.storage**: The storage size of the volume. • **volume.createOption**: By default, the dm-prov-telco pod uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **config.totalFrontEnds**: The total number of front end processes. The default is **4**. • **config.totalBackEnds**: The total number of back end processes. The default is **8**. • **config.connectionsPerFrontEnd**: The number of connections for each front end process. The default is **16**. • **config.connectionsPerBackEnd**: The number of connections for each back end process. The default is **8**. • **config.sharedMemoryBigSize**: The size of shared memory for large shared-memory structures. The default is **2097152**. • **config.sharedMemorySize**: The amount of shared memory allocated for use by Provisioning DM processes. The default is **33554432**. • **resources.***: The minimum and maximum CPU and memory resources for the pod. • **nodeSelector**: The rules for deploying the Provisioning DM pod on specific nodes. • **affinity**: The rules for running the Provisioning DM pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.
storage_class.*	**ocbrm**	The details for the Kubernetes StorageClass used by BRM: • **create**: Whether to create a Kubernetes StorageClass (**true**) or use an existing StorageClass (**false**). The default is **true**. • **name**: The name of the Kubernetes StorageClass to create or use. The default is **green**. • **parameters**: The provisioner-specific parameters for the StorageClass. The supported parameters depend on the storage provisioner. By default, no parameters are specified.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
smtpServer	**ocbrm.dm_email.deployment**	The SMTP server name. The default is **example.us.oracle.com**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
db.*	**ocbrm**	The details for connecting to the BRM database: • **deployment.***: The details for deploying the BRM database initialization pod, such as the image name and image tag. • **host**: The host name or IP address of the database server. • **port**: The port number of the database server. • **service**: The service name that identifies the database. • **sslMode**: The type of SSL connection required for connecting to the database: **TWO_WAY, ONE_WAY**, or **NO**. • **extDBSSLWalletSecret**: The name of the Kubernetes Secret containing the SSL database wallet. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **walletPassword**: The password for accessing the certificates from the TrustStore and KeyStore. This is required if **sslMode** is set to **ONE_WAY** or **TWO_WAY**. • **walletType**: The wallet type for database SSL connections: **sso** or **pkcs12**. The default is **sso**. • **sslServerCertDN**: The database SSL server certificate DN for components that connect through JDBC. • **tnsAdmin.connectionDescriptor**: Use this key when the database connection requires a RAC address list or another custom descriptor. The chart adds the **pindb** alias. When this value is empty, **oc-cn-helm-chart** generates the existing connection descriptor from the database host, port, service name, and SSL settings. • **tnsAdmin.sqlnet**: Set this to the complete **sqlnet.ora** configuration for the BRM database. When this value is empty, **oc-cn-helm-chart** generates the original SQLnet configuration. • **enable_partition**: Whether partitioning is enabled at the database level (**Yes**) or disabled (**No**). The default is **Yes**. • **enable_delayed_partition**: Whether delayed partitioning is enabled for all subclasses of the **/event** storable class (**Yes**) or not (**No**). The default is **Yes**. • **storage_model**: The size of the BRM database tablespaces: – **Test**: Less than 700 MB. – **Small**: Less than 1.5 GB. – **Medium**: Less than 30 GB. – **Large**: Greater than 30 GB. • **schemauser**: The user name of the primary BRM schema. The default is **pin**. • **schemapass**: The password for the BRM schema. • **schematablespace**: The name of the tablespace for the primary BRM schema. This field is case-sensitive. The default is **pin**. • **indextablespace**: The name of the index tablespace for the primary BRM schema. This field is case-sensitive. The default is **pinx**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **nls_lang**: The language, territory, and character set. The default is **AMERICAN_AMERICA.AL32UTF8**. • **pipelineschemauser**: The BRM pipeline schema user name, which should be pre-created with all of the required grants. • **pipelineschemapass**: The BRM pipeline schema password. You must generate a Secret. Add this key to your **override-values.yaml** file with the same path hierarchy. • **pipelineschematablespace**: The name of the tablespace for the BRM pipeline schema. This field is case-sensitive. • **pipelineindextablespace**: The name of the index tablespace for the BRM pipeline schema. This field is case-sensitive. • **skipPrimary**: Whether to deploy the primary schema (**false**) or skip the deployment of the primary schema (**true**). Set it to **true** only if you are adding a schema to an existing BRM multischema system. The default is **false**. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **process_online_indexes**: Whether to create indexes online for tables with large amounts of data. The default is **NO**. Ensure these values match the **ocbrm.db.*** keys from **oc-cn-init-db-helm-chart**. See Table 6-1 for more information.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description

secondary*N*.*	**ocbrm.db.multiSchemas**	The details for connecting to your secondary database schemas, where *N* is **1** for the first secondary schema, **2** for the next secondary schema, and so on.
		Add this block only if your BRM database contains multiple schemas. This section will be commented out by default:
		
		`multiSchemas:`
		`  secondary:`
		`  deploy:`
		`  host: localhost`
		`  port: 1521`
		`  service: pindb`
		`  schemauser:`
		`  schemapass:`
		`  schematablespace:`
		`  indextablespace:`
		`  tnsAdmin:`
		`  connectionDescriptor:`
		
		See "Configuring BRM for a Multischema Database".

upgrade.*	**ocbrm**	The details for running the BRM database upgrade: • **deployment.***: The details for the upgrade image. • **resources.***: The minimum and maximum CPU and memory resources for the upgrade pod. • **nodeSelector**: The rules for deploying the upgrade pod on specific nodes. • **affinity**: The rules for running the upgrade pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.
perflib.*	**ocbrm**	The details for configuring the performance library: • **deployment.imageName**: The name of the performance library container image. The default is **perflib**. • **deployment.imageTag**: The tag for the performance library container image. • **deployment.persistPerflibLogs**: Whether to store performance library timing files in the BRM database. The default is **false**. • **deployment.jvmOpts**: The Java options for the performance library JVM. • **deployment.perfdOptions**: The options for the performance daemon. The default is **-A -P120d**. • **resources.***: The minimum and maximum CPU and memory resources for the pod.
provisioner	**ocbrm.dynamic_provisioner**	The name of the external storage provisioner. The default is **oracle.com/oci**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
dm_fusa.*	**ocbrm**	The details for configuring the Paymentech DM: • **isEnabled**: Whether to enable the Paymentech DM. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **fusaSimulator.extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **batchProto**: The connection type between **dm_fusa** and Paymentech for batch payment transactions: TCP/IP (**socket**), SFTP (**sftp**), or the connection is offline (**linkdown**). The default is **socket**. • **deployment.***: The details for deploying the Paymentech DM. • **volume.storage**: The storage size of the volume. • **volume.createOption**: By default, the dm-fusa pod uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the **dm_fusa** pod on specific nodes. • **affinity**: The rules for running the **dm_fusa** pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **secret.extDmFusaPasswordSecret**: The name of the external Kubernetes Secret containing the passwords for the Paymentech DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **secret.extPvtKeySecret**: The name of the external Kubernetes Secret containing the private key for the Paymentech DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **secret.extPubKeySecret**: The name of the external Kubernetes Secret containing the public key for the

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		Paymentech DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **secret.extPvtKeyPpSecret**: The name of the external Kubernetes Secret containing the private key passcode for the Paymentech DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **dnRecord**: The distinguished-name record setting for the Paymentech DM. • **appendResponse**: The setting that controls whether the Paymentech DM appends response data. • **sftpSshConfig.***: The SSH connection settings for SFTP transfers. • **sftpParams.***: The SFTP directory and retry settings for the Paymentech DM. • **secret.sftpKey**: The name or value used for the SFTP private key Secret. • **secret.sftpKeyPassPhrase**: The SFTP private key passphrase Secret. • **secret.sftpAuthorizedKeys**: The SFTP authorized-keys Secret. • **secret.sftpPwd**: The SFTP password Secret.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
dm_vertex.*	**ocbrm**	The details for configuring the Vertex DM: • **isEnabled**: Whether to enable the Vertex DM. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the Vertex DM. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **secret.extDmVertexPasswordSecret**: The name of the external Kubernetes Secret containing the KeyStore passwords for the Vertex DM. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
taxation_gateway.*	**ocbrm**	The details for configuring the taxation gateway. • **isEnabled**: Whether to enable the Taxation Gateway. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the taxation gateway. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **secret.client_id**: The client ID generated by the Identity Server, used to validate the OAuth 2.0 token. • **secret.client_secret**: The Base64-encoded IDCS client secret. • **secret.extTaxationGatewaySecret**: The name of the external Kubernetes Secret containing the KeyStore passwords for the taxation gateway. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
rule_manager.*	**ocbrm**	The details for configuring Rule Manager: • **isEnabled**: Whether to enable Rule Manager. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes Secret that contains the component-specific TLS wallet. • **deployment.***: The details for deploying the Rule Manager pod. • **configEnv.ruleFilesRefreshIntervalSeconds**: The interval, in seconds, for refreshing the rule files. The default is **60**. • **resources.***: The minimum and maximum CPU and memory resources for the pod. • **nodeSelector**: The rules for deploying the Rule Manager pod on specific nodes. • **affinity**: The rules for running the Rule Manager pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
realtimepipe.*	**ocbrm**	The details for configuring the Realtime Pipeline: • **isEnabled**: Whether to enable the Realtime Pipeline (**true**) or not (**false**). The default is **true**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.replicaCount**: The number of replicas to create of the realtimepipe pod. The default is **1**. • **deployment.rtp_num_thread**: The number of threads in the realtime pipeline. The default is **8**. • **deployment.rtp_num_pipe**: The number of realtime pipelines. The default is **2**. • **deployment.discount_trace**: Whether to generate a discount trace file. The default is **true**. • **deployment.SemaphoreEnable**: Whether to check for and process semaphore files, which allow you to configure and control Pipeline Manager during runtime. The default is **true**. For more information, see "Using Semaphore Files to Control Pipeline Manager" in *BRM Pipeline Manager Administration Guide*. • **volume.storage**: The storage size of the volume. • **volume.createOption**: By default, the realtimepipe pod uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **deployment.imageName**: The name of the realtime pipeline container image. The default is **realtimepipe**. • **deployment.imageTag**: The tag for the realtime pipeline container image. • **deployment.console_discount_trace**: Whether to write discount trace information to the console. The default is **True**. • **deployment.loglevel**: The logging level for the realtime pipeline. The default is **major**. • **nodeSelector**: The rules for deploying the realtime pipeline pod on specific nodes. • **affinity**: The rules for running the realtime pipeline pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. • **resources.***: The minimum and maximum CPU and memory resources for the pod. • **hpaValues.***: The details for scaling the number of pod replicas based on CPU or memory utilization.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
batch_controller.*	**ocbrm**	The details for configuring the Batch Controller: • **isEnabled**: Whether to enable the BRM Controller. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the Batch Controller. • **volume.***directory***.createOption**: By default, all of the pipelines in the batch-controller pod uses dynamic volumes. To use a static volume instead, you must add the **createOption** key under the *directory* key, where *directory* is **input**, **archive**, and **reject**. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **volume.directory.storage**: The storage size of the volume, where directory is input, archive, or reject. • **nodeSelector**: The rules for deploying the Batch Controller pod on specific nodes. • **affinity**: The rules for running the Batch Controller pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. • **resources.***: The minimum and maximum CPU and memory resources for the pod. • **hpaValues.***: The details for scaling the number of pod replicas based on CPU or memory utilization.
dm_eai.*	**ocbrm**	The details for configuring EAI DM: • **isEnabled**: Whether to enable EAI DM. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes Secret that contains the component-specific TLS wallet. • **deployment.***: The details for deploying the EAI DM pod. • **resources.***: The minimum and maximum CPU and memory resources for the pod. • **nodeSelector**: The rules for deploying the EAI DM pod on specific nodes. • **affinity**: The rules for running the EAI DM pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. • **hpaValues.***: The details for scaling the number of pod replicas based on CPU or memory utilization.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
rel_daemon.*	**ocbrm**	The details for configuring the Rated Event (RE) Loader daemon: • **isEnabled**: Whether to enable the RE Loader daemon. The default is **true**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **job.isEnabled**: Whether to run the RE Loader daemon job. The default is **false**. • **job.extConfigScriptsCM**: The name of the ConfigMap for the external configuration scripts. • **deployment.***: The details for deploying the RE Loader daemon. • **volume.***directory***.createOption**: By default, all of the directories in the rel-daemon pod use dynamic volumes. To use static volumes instead, you must add the **createOption** key under the *directory* key, where *directory* is **reject**, **archive**, and **input**. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **hpaValues.***: The details for scaling up or down the number of pod replicas in your deployment based on the pod's CPU or memory utilization. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*. • **job.dbNumber**: The database number for the RE Loader daemon job. The default is **0.0.0.1**. • **job.jvmOpts**: The JVM options to use when running the RE Loader daemon job. • **job.nodeSelector**: The rules for deploying the RE Loader daemon job on specific nodes. • **job.affinity**: The rules for running the RE Loader daemon job on specific nodes.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **job.addOnPodSpec**: The details for extending the job pod specification or overriding features. By default, this key is empty. • **volume.directory.storage**: The storage size of the volume, where directory is reject, archive, or input.
rem.*	**ocbrm**	The details for configuring the rated-event-manager pod. **Note**: If **ocbrm.db.sslMode** is enabled (**ONE_WAY** or **TWO_WAY**), you must specify the server certificate's distinguished name (DN) in the **oc-cn-helm-chart/templates/configmap_rem_properties.yaml** file's **rated_event_manager.jdbc_pool.sslServerCertDN** key. • **logging.***: The details for logging information about the rated-event-manager pod. • **isEnabled**: Whether to deploy the rated-event-manager pod. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the rated-event-manager pod, such as its image name, its image tag, and the Kafka topic to use for rated events. • **volume.***directory***.createOption**: By default, all of the directories in this pod use dynamic volumes. To use a static volume instead, you must add the **createOption** key under the *directory* key, where *directory* is **input**, **reject**, **archive**, and **data**. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **volume.***directory***.storage**: The storage size of the volume. • **nodeSelector**: The rules for deploying the rated-event-manager pod on specific nodes. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **hpaValues.***: The details for scaling up or down the number of pod replicas in your deployment based on the pod's CPU or memory utilization. By default, the Horizontal Pod Autoscaler is disabled. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*. • **affinity**: The rules for running the rated-event-manager pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. • **healthCheckPort**: The port used for health checks. The default is **8989**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
rel_manager.*	**ocbrm**	The details for configuring the RE Loader job. • **extUserSSLWalletSecret**: The name of the Kubernetes Secret containing the component-specific SSL/TLS wallet, including the user certificate and private key used for TLS authentication. • **job.extConfigScriptsCM**: The name of the external ConfigMap containing the configuration scripts to run. • **job.isEnabled**: Whether to enable the RE Loader Manager job. The default is **false**. • **job.dbNumber**: The database number for the RE Loader job. The default is **0.0.0.1**. • **job.jvmOpts**: The Java Virtual Machine (JVM) options to use when running the RE Loader Manager job, such as initial and maximum heap size, container support, and other JVM system properties. • **job.logging.***: The details for logging information about the pod. • **job.resources**: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **job.nodeSelector**: The rules for deploying the pod on specific nodes. • **job.affinity**: The rules for running the rated-event-manager pod on specific nodes. • **job.addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
event_stream_processor.*	**ocbrm**	The details for configuring the event stream processor, which is used to load prerated events into the BRM database. **Note**: If **ocbrm.db.sslMode** is enabled (**ONE_WAY** or **TWO_WAY**), you must specify the server certificate's distinguished name (DN) in the **oc-cn-helm-chart/templates/configmap_esp_properties.yaml** file's **event_stream_processor.jdbc_pool.sslServerCertDN** key. • **isEnabled**: Whether to deploy the event stream processor pod. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **extEspCustomSpecsCM**: The name of the external ConfigMap containing custom specifications for the event stream processor. • **deployment.***: The details for deploying the event stream processor, such as its image name, its image tag, and replica count. • **configEnv.port**: The port used by the event stream processor service. The default is **8080**. • **configEnv.metricsMonitorThreadName**: The name of the thread that monitors Kafka consumer metrics. The default is **kafka-consumer-metrics-monitor**. • **configEnv.metricsMonitorRetryInterval**: The retry interval used by the Kafka consumer metrics monitor. The default is **6**. • **configEnv.kafkaBootstrapServer**: The address of the Kafka bootstrap server that the event stream processor uses to connect to Kafka. • **configEnv.kafkaAutoOffsetReset**: The default is **earliest**. • **configEnv.kafkaKeyDeserializer**: The deserializer class used for Kafka message keys. The default is **org.apache.kafka.common.serialization.StringDeserializer**. • **configEnv.kafkaValueDeserializer**: The deserializer class used for Kafka message values. The default is **io.cloudevents.kafka.CloudEventDeserializer**. • **configEnv.kafkaMaxPollRecords**: The maximum number of records returned in a single Kafka poll. The default is **5**. • **configEnv.kafkaServerCheckRetries**: The number of attempts to connect to the Kafka server if it is unreachable. The default is **5**. • **configEnv.kafkaServerCheckTimeout**: The time, in milliseconds, to wait for a response from the Kafka server for each connection attempt. The default is **10000**. • **configEnv.kafkaServerBatchTimeout**: The maximum time, in milliseconds, to wait for a batch of messages from the Kafka server. The default is **30000**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **configEnv.kafkaServerPollTimeout**: The maximum time, in milliseconds, to wait for messages during each Kafka poll. The default is **1000**. • **configEnv.defaultEventHandler**: The Java class that handles events when a topic-specific event handler is not configured. The default is **com.oracle.brm.esp.core.event_stream_processor.handler.EventHandlerImpl**. • **configEnv.isSecurityEnabled**: Whether security is enabled for connections between the event stream processor and Kafka. The default is **false**. • **configEnv.isClientKeystoreEnabled**: Whether the client KeyStore is enabled for secure Kafka connections. The default is **false**. • **configEnv.securityProtocol**: The security protocol used for connections between the event stream processor and Kafka. • **configEnv.saslMechanism**: The SASL authentication mechanism used to connect to Kafka. • **configEnv.saslLoginModule**: The SASL login module used for Kafka authentication. • **configEnv.saslUsername**: The user name used for SASL authentication with Kafka. • **configEnv.trustStorePassword**: The password for the TrustStore used for secure Kafka connections. • **configEnv.keyStorePassword**: The password for the KeyStore used for secure Kafka connections. • **configEnv.keyPassword**: The password for the key in the KeyStore used for secure Kafka connections. • **configEnv.topicGroups.***: The name of the Kafka topic from which the Event stream processor application consumes rated events. • **secret.extOmsSchemaPasswordSecret**: The name of the external Kubernetes Secret containing the passwords for the OMS database schema. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **logging.***: The details for logging information about the event stream processor pod. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **configEnv.password**: The password used for Kafka authentication. • **configEnv.clientId**: The client ID used for Kafka authentication. • **configEnv.clientSecret**: The client secret used for Kafka authentication. • **configEnv.espPartitionSet**: The partition-set value used by the event stream processor. The default is **1**. • **configEnv.espPoidIncrementBy**: The value by which the event stream processor increments POIDs. The default is **5**. • **configEnv.espRefBatchingEnabled**: Whether reference batching is enabled. The default is **false**. • **configEnv.espTimeSource**: The time source used by the event stream processor. The default is **system**. • **configEnv.espDedupEnabled**: Whether event deduplication is enabled. The default is **true**. • **configEnv.espDedupRetentionDays**: The number of days to retain event deduplication data. The default is **30**. • **configEnv.espDedupSprocLoggingEnabled**: The logging setting for event-deduplication stored procedures. The default is **-1**. • **configEnv.jvmOpts**: The Java options for the event stream processor JVM.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
batchpipe.*	**ocbrm**	The details for configuring the batch pipeline. • **isEnabled**: Whether to enable the batch pipeline. The default is **true**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the batch pipeline. • **volume.output.storage**: The storage size of the volume. • **volume.*directory*.createOption**: By default, all of the directories in the batchpipe pod use dynamic volumes. To use a static volume instead, you must add the **createOption** key under the *directory* key, where *directory* is **data**, **output**, **reject**, and **log**. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **volume.directory.storage**: The storage size of the volume, where directory is data, output, reject, or log. • **nodeSelector**: The rules for deploying the batch pipeline pod on specific nodes. • **affinity**: The rules for running the batch pipeline pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
roampipe.*	**ocbrm**	The details for configuring the roaming pipeline: **Note:** The roampipe pod is a single-replica pod and must not be scaled to multiple replicas that share a file system. To provide high availability, deploy separate roampipe pods, each with its own file system. Route incoming CDR files so that each pod processes a distinct set of files. This configuration prevents multiple pods from processing the same file. • **isEnabled**: Whether to enable the roaming pipeline. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **deployment.***: The details for deploying the roaming pipeline. • **volume.***directory***.createOption**: By default, all of the pipelines in the roampipe pod use dynamic volumes. To use a static volume instead, you must add the **createOption** key under the *directory* key, where *directory* is **output** and **reject**. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **volume.directory.storage**: The storage size of the volume, where directory is output or reject. • **nodeSelector**: The rules for deploying the roaming pipeline pod on specific nodes. • **affinity**: The rules for running the roaming pipeline pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. • **resources.***: The minimum and maximum CPU and memory resources for the pod.
formatter.*	**ocbrm**	The details for configuring the Formatter: • **isEnabled**: Whether to enable the Formatter. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes Secret that contains the component-specific TLS wallet. • **deployment.***: The details for deploying the Formatter pod. • **resources.***: The minimum and maximum CPU and memory resources for the pod. • **nodeSelector**: The rules for deploying the Formatter pod on specific nodes. • **affinity**: The rules for running the Formatter pod on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
cmt.*	**ocbrm**	The details for configuring and running **pin_cmt**. • **enabled**: Set this to **true** to run the Conversion Manager **pin_cmt** utility. The default is **false**. • **volume.storage**: The storage size of the volume. • **volume.createOption**: By default, the cmt pod uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.
config_jobs.*	**ocbrm**	The details for running a configurator job, which allows you to run BRM load utilities on demand without entering into a pod: • **deployment.***: The details for deploying the configurator job. • **run_apps**: Set to **true** to enable a configurator job. The default is **false**. • **isMultiSchema**: Specifies whether to run the commands in the **loadme.sh** script on the secondary schemas. The default is **false**. • **restart_count**: Increment this value by 1 to restart the CM during a Helm upgrade. The default is **0**. • **script_name**: The name of the script that contains the load utilities you want to run. The default is **loadme.sh**. • **configmap_path**: The directory in which ConfigMaps are stored. The default is **/oms/load**. • **extCustomScriptsCM**: The name of the ConfigMap containing custom scripts to run. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the configurator job on specific nodes. • **affinity**: The rules for running the configurator job on specific nodes. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "Running Load Utilities through Configurator Jobs" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
brm_apps.*	**ocbrm**	The details for running a brm-apps job, which allows you to run BRM applications and utilities on demand without entering into a pod: • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **job.isEnabled**: Whether to enable a brm-apps job. The default is **false**. • **job.isMultiSchema**: Specifies whether to run the commands in the **loadme.sh** script on the secondary schemas. The default is **false**. • **job.configmap_path**: The directory in which the ConfigMap is mounted. The default is **/oms/load**. • **job.script_name**: The name of the script that contains the utilities and applications you want to run. The default is **loadme.sh**. • **extCustomScriptsCM**: The name of the ConfigMap containing custom scripts to run. • **deployment.***utilityName***.***: The configuration details for running BRM utilities and applications, such as **pin_billd**, **pin_export_price**, and **pin_rerate**. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **hpaValues.***: The details for scaling up or down the number of pod replicas in your deployment based on the pod's CPU or memory utilization. By default, the Horizontal Pod Autoscaler is disabled. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*. • **secret.extOmsBrmAppsPasswordSecret**: The name of the external Kubernetes Secret containing the passwords for the brm-apps job. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **job.jvmOpts**: The Java options for the BRM Apps job JVM. • **job.nodeSelector**: The rules for deploying the BRM Apps job on specific nodes. • **job.affinity**: The rules for running the BRM Apps job on specific nodes. • **job.addOnPodSpec**: The details for extending the job pod specification or overriding features. By default, this key is empty. • **deployment.isEnabled**: Whether to enable the BRM Apps deployment. The default is **true**. • **deployment.imageName**: The name of the BRM Apps container image. The default is **brm_apps**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **deployment.imageTag**: The tag for the BRM Apps container image. • **deployment.replicaCount**: The number of BRM Apps pod replicas to create. The default is **1**. • **deployment.nodeSelector**: The rules for deploying BRM Apps pods on specific nodes. • **deployment.affinity**: The rules for running BRM Apps pods on specific nodes. • **deployment.addOnPodSpec**: The details for extending the deployment pod specification or overriding features. By default, this key is empty. • **deployment.jvmOpts**: The Java options for the BRM Apps deployment JVM. See "Running Applications and Utilities through BRM-Apps Jobs" in *BRM Cloud Native System Administrator's Guide*.
extUserSSLWallet Secret	**ocbrm.wsm**	The name of the Kubernetes Secret that contains the component-specific TLS wallet for Web Services Manager.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
wsm.soap.*	**ocbrm**	Details about the Web Services Manager service. • **isEnabled**: Whether to enable SOAP-based Web Services Manager services. The default is **false**. • **deployment.***: The details for deploying the brm-wsm pod. • **service.type**: The service type. The default is **ClusterIP**. • **service.resources.***: The minimum and maximum CPU and memory resources that brm-wsm can use. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **configEnv.port**: The HTTP port where Web Services Manager is exposed. The default is **8080**. • **configEnv.httpsPort**: The HTTPS port where Web Services Manager is exposed. The default is **8443**. • **configEnv.inputValidationEnabled**: Whether to validate the input XML payload. The default is **true**. • **configEnv.soapInputValidationReportOnly**: Whether the validation errors in the input XML payload are logged to the console (**true**) or sent back as a fault only (**false**). The default is **false**. • **configEnv.outputValidationEnabled**: Whether to validate the output XML payload. The default is **true**. • **configEnv.soapOutputValidationReportOnly**: Whether the validation errors in the output XML payload are logged to the console (**true**) or sent back as a fault only (**false**). The default is **false**. • **configEnv.logLevel**: The logging level. The default is **WARNING**. • **configEnv.tlsEnabled**: Whether to enable TLS encryption for Web Services Manager. The default is **false**. • **configEnv.externalSecretName**: The name of the Kubernetes Secret containing the passwords for Web Services Manager. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **configEnv.clientAuth**: The client-authentication mode. The default is **OPTIONAL**. • **configEnv.jvmOpts**: The Java options to configure, such as heap memory and JVM configurations. • **configEnv.keyStoreFileName**: The file name of the KeyStore. • **configEnv.keyStoreAlias**: The private key alias of the KeyStore. • **configEnv.trustStoreFileName**: The file name of the TrustStore. • **configEnv.isOauthEnabled**: Whether OAuth 2.0 is enabled in Web Services Manager. The default is **false**. • **configEnv.oauthCertificateName**: The name of the OAuth certificate file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
		• **configEnv.outputDateFormat**: The date and time format used for the output XML payload file name. The default is **yyyy-MM-dd'T'HH:mm:ss**. • **configEnv.outputPrefixUnixTimestamp**: Whether to add a UNIX time stamp prefix to the output XML payload file name. The default is **false**. • **configEnv.inputDateFormat**: The date and time format used for the input XML payload file name. The default is **yyyy-MM-dd'T'HH:mm:ss**. • **configEnv.inputPrefixUnixTimestamp**: Whether to add a UNIX time stamp prefix to the input XML payload file name. The default is **false**. • **configEnv.outputNamespacePrefixSoap**: The prefix to add to SOAP XML elements in the response. The default is **S**. • **configEnv.outputNamespacePrefixPayload**: The prefix to add to the payload XML elements in the response. The default is **brm**. • **configEnv.tracing.***: The details for tracing the BRM-WSM service. See "Using Metrics and Tracing" in *BRM Web Services Manager*. • **secrets.***: The name of the Secrets containing the KeyStore and TrustStore passwords. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*. • **configEnv.extKeystoreSecret**: The name of the Kubernetes Secret containing the Web Services Manager keystore. For more information about Web Services Manager, see "Using Web Services" in *BRM Web Services Manager*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
brm_sdk.*	**ocbrm**	The details for the brm-sdk pod: • **isEnabled**: Whether to enable the BRM SDK. The default is **false**. • **extUserSSLWalletSecret**: The name of the Kubernetes secret that contains the component-specific TLS wallet, including the user certificate and private key used for TLS authentication. See "Configuring TLS Certificates for BRM Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **extCustomScriptsCM**: The name of the external ConfigMap containing your custom scripts. • **deployment.***: The details for deploying the BRM SDK. • **volume.storage**: The storage size of the volume. • **volume.createOption**: By default, the brm-sdk pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **resources.***: The minimum and maximum CPU and memory resources for the pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the pod on specific nodes. • **affinity**: The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. • **addOnPodSpec**: The details for extending the pod specification or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.
logging.*	**N/A**	The details for forwarding logs to a log management system: • **isEnabled**: Whether log forwarding is enabled. The default is **true**. • **fluentd.imageName**: The name of the Fluentd container image. • **fluentd.resources.***: The minimum and maximum CPU and memory resources for the Fluentd container. • **fluentd.conf.***: The component-specific Fluentd configuration.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-5 (Cont.) BRM Server Keys

Key	Path in values.yaml File	Description
webhook.*	N/A	The details for the webhook pod: • **isEnabled**: Whether to enable the webhook application. The default is **false**. • **logPath**: The path to write log files for the webhook application. • **logLevel**: The logging level. The default is **INFO**. • **extScriptsCM**: The name of the external ConfigMap containing the scripts to run. • **deployment.***: The details for deploying the webhook pod. • **scripts.mountPath**: The directory in which you store any scripts to be run by the webhook application. The default is **/u01/script**. • **wop.***: The details for WebLogic Operator. • **jsonConfig**: (Optional) Extra alert details in JSON format. For more information, see "Configuring webhook to Enable Autoscaling" and "Configuring webhook to Enable Autoscaling (PCC)" in *BRM Cloud Native System Administrator's Guide*.

Configuring BRM for a Multischema Database

Using a BRM multischema database lets you distribute customer accounts among several database schemas, providing increased storage capacity, higher performance, and easier maintenance. For more information, see "A BRM Multischema Production System" in *BRM Installation Guide*.

To configure your BRM cloud native environment to connect to a multischema database, do this:

1. Ensure that you deployed a multischema database in your BRM cloud native environment. See "Deploying BRM with a New Database Schema".
2. Using SQL*Plus, grant each database schema the privilege to insert and update tables on the other schemas:
   a. Connect to the BRM database with SQL*Plus as **sysdba**:

      ```
      sqlplus system@databaseAlias as sysdba
      Enter password: password
      ```

      where:

      - *databaseAlias* is the Oracle system database alias.
      - *password* is the Oracle system database user password.

   b. From the primary database schema, enter the following:

      ```
      SQL> GRANT INSERT ANY TABLE TO SecondarySchema;
      SQL> GRANT UPDATE ANY TABLE TO SecondarySchema;
      ```

      where *SecondarySchema* is the name of the secondary schema.

Copyright © 2019, 2026, Oracle and/or its affiliates.

c. From each secondary database schema, enter the following:

```
SQL> GRANT INSERT ANY TABLE TO PrimarySchema;
SQL> GRANT UPDATE ANY TABLE TO PrimarySchema;
```

where *PrimarySchema* is the name of the primary schema.

3. Connect the BRM server to each secondary schema:
   a. Open your **override-values.yaml** file for **oc-cn-helm-chart**.
   b. Enable account migration by setting the **ocbrm.isAmt** key to **true**.
   c. Set the **ocbrm.db.skipPrimary** key to **false**.
   d. For each secondary schema in your system, add a **ocbrm.db.multiSchemas.secondary***N* block, where *N* is **1** for the first secondary schema, **2** for the next secondary schema, and so on.
   e. In each **ocbrm.db.multiSchemas.secondary***N* block, set the following keys:
      - **deploy**: Set this to **true**.
      - **host**: Set this to the hostname of the secondary schema. This key is optional.
      - **port**: Set this to the port number for the secondary schema. This key is optional.
      - **service**: Set this to the service name for the secondary schema. This key is optional.
      - **schemauser**: Set this to the schema user name.
      - **schemapass**: Set this to the schema password.
      - **schematablespace**: Set this to the name of the schema tablespace, such as pin01.
      - **indextablespace**: Set this to the name of the index tablespace, such as pinx01.
      - **tnsAdmin.connectionDescriptor**: Set this to a custom Oracle Net connection descriptor for the secondary schema. When you set this key, **oc-cn-helm-chart** instead of generating one from the host, port, and service keys. This key is optional.
   f. Deploy **oc-cn-helm-chart** by running this command from the **helmcharts** directory:

      ```
      helm install BrmReleaseName oc-cn-helm-chart --namespace BrmNameSpace --values
      OverrideValuesFile
      ```

      where:
      - *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance. It must be different from the one used for **oc-cn-init-db-helm-chart**.
      - *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.
      - *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-helm-chart**.

      The BRM Helm chart deploys new dm-oracle, amt, and rel-daemon pods, Rated Event (RE) Loader PVCs, services, ConfigMaps, and secrets. It also updates their corresponding schema entries in the primary CM and Oracle DM and deploys multiple containers for the batch-wireless-pipe pod.

Copyright © 2019, 2026, Oracle and/or its affiliates.

4. Set each database schema's status and priority. BRM cloud native assigns accounts to an open schema with the highest priority.

   a. Open the **configmap_pin_conf_testnap.yaml** file.

   b. Under the **config_dist.conf** section, add the following entries for each secondary schema in your database:

   ```
   DB_NO = "schema_number" ;             # database config. block
   PRIORITY = priority ;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "status" ;
   SCHEMA_NAME = "schema_name" ;
   ```

   c. Set the **STATUS** and **PRIORITY** entries for each primary and secondary schema:

   ```
   DB_NO = "0.0.0.1" ;             # Primary schema configuration block
   PRIORITY = priority;
   MAX_ACCOUNT_SIZE = 100000 ;
   STATUS = "status" ;
   SCHEMA_NAME = "schema_name" ;

   DB_NO = "0.0.0.2" ;             # Secondary schema configuration block
   PRIORITY = priority;
   MAX_ACCOUNT_SIZE = 50000 ;
   STATUS = "status" ;
   SCHEMA_NAME = "schema_name" ;
   ```

   where:

   - *priority* is a number representing the schema's priority, with the highest number having the most priority. For example, 5 indicates a greater priority than a value of 1. For more information, see "Modifying Database Schema Priorities" in *BRM Cloud Native System Administrator's Guide*.
   - *status* specifies whether the schema is **open**, **closed**, or **unavailable**. For more information, see "Modifying Database Schema Status" in *BRM Cloud Native System Administrator's Guide*.

   d. Set up the configurator job to run the **load_config_dist** utility by adding the following lines to the **oc-cn-helm-chart/config_scripts/loadme.sh** script:

   ```
   #!/bin/sh

   #cp /oms/config_dist.conf /oms/sys/test/config_dist.conf
   cd /oms/sys/test ; load_config_dist
   exit 0;
   ```

   e. In the **override-values.yaml** file for **oc-cn-helm-chart**, set this key:

   ocbrm.config_jobs.run_apps: Set this to **true**.

   f. Run the **helm upgrade** command to update the Helm release:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --
   namespace BrmNameSpace
   ```

   The distribution information is loaded into the primary schema.

   g. Update these keys in the **override-values.yaml** file for **oc-cn-helm-chart**:

   - **ocbrm.config_jobs.restart_count**: Increment the existing value by 1.
   - **ocbrm.config_jobs.run_apps**: Set this to **false**.

   h. Update the Helm release again:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --
namespace BrmNameSpace
```

The CM pod is bounced back.

5. Configure the account-router Pipeline Manager to route CDRs to pipelines based on the database schema POID. To do so, edit the ConfigMap file **configmap_acc_router_reg.yaml**.

   Based on the configuration, the account router Pipeline Manager does the following:

   - Moves input files to the **data** PVC directory. The input file names have a prefix of **router** and a suffix of **.edr**.
   - Moves the rated output files to the input of the Rating pipeline.
   - Replicates the Rating pipeline based on the multischema entry. The Range function is used to replicate the rating pipeline.
   - Moves the output files from the Rating pipeline to the **outputcdr** PVC directory.

Your BRM cloud native environment is connected to your BRM multischema database. To manage your multischema environment, see "Managing a BRM Cloud Native Multischema System" in *BRM Cloud Native System Administrator's Guide*.

Configuring Pricing Design Center

Pricing Design Center (PDC) is a Web-based client application that you use to create and manage the product offerings that you sell to your customers. A product offering represents the services available to your customers and the price of those services. For more information about PDC, see *Pricing Design Center Online Help*.

You can optionally deploy a simple demonstration version of Pricing Design Center cloud native by using the sample **PDC_OverrideValues.yaml** file provided with **oc-cn-helm-chart**. This simple demonstration version has both SSL and ECE enabled, loads sample RUMs and balance elements, and starts the BRM-to-PDC synchronization process with **SyncPDC**.

To configure PDC to run in your BRM cloud native environment:

1. Override the PDC-specific keys in the **values.yaml** file for **oc-cn-helm-chart**. See "Adding PDC Keys for oc-cn-helm-chart".
2. Override the PDC-specific keys in the **values.yaml** file for **oc-cn-op-job-helm-chart**. See "Adding PDC Keys for oc-cn-op-job-helm-chart".
3. Set up SAML for SSO in PDC. See "Setting Up SSO for PDC Cloud Native".

After you deploy PDC in your cloud native environment, you can access the PDC UI at one of the following URLs:

- `http://kubernetesHost:pdcPort/pdc`

  where:

  - *kubernetesHost* is the host name of the machine on which Kubernetes is deployed.
  - *pdcPort* is the PDC service node port.

- `http://LoadBalancerHost:pdcNodePort/pdc`

  where:

  - *LoadBalancerHost* is the host name of the machine on which the load balancer is deployed.

Copyright © 2019, 2026, Oracle and/or its affiliates.

– *pdcNodePort* is the number assigned to the PDC node port.

Adding PDC Keys for oc-cn-helm-chart

Table 7-6 describes the most common PDC keys that you need to override. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart** with the same path hierarchy.

For information about all PDC-specific keys, see the descriptions in the **oc-cn-helm-chart/values.yaml** file.

> ⚠ **Caution**
>
> Keys with the path **ocpdc.secretValue** hold sensitive data. Handle them carefully with controlled access to the file containing their values. Encode all of these values in Base64. See "Secrets" in *Kubernetes Concepts*.

Table 7-6 Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
isEnabled	**ocpdc**	Whether to enable and deploy PDC: • **true**: Enables PDC and deploys the PDC application. This is the default. • **false**: Disables the PDC application.
lang	**ocpdc**	The Linux system locale. The default is **en_US.UTF-8**.
tz	**ocpdc**	The Linux time zone. The default is **UTC**.
volume.createOption.*	**ocpdc**	By default, PDC uses dynamic volume provisioning. PDC on WebLogic Kubernetes Operator requires shared storage for the WebLogic domain home and configuration in clustered deployments. Use a storage class that supports ReadWriteMany, or use an equivalent shared storage solution such as NFS. To use a static volume instead, add the **createOption** keys under the **volume** section. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. **Note**: Ensure you provide the required permissions to the volume path by following the guidelines in "Persistent Volume Storage Locations" in the WebLogic Kubernetes Operator documentation.
storageSize	**ocpdc**	The storage size for the pdc-brm-pv or pdc-brm-pvc, such as 10 Gi. The storage must use an access mode supported for PDC shared storage in your deployment.
deployment.*	**ocpdc**	The details for deploying the PDC image: • **imageName**: The name of the PDC image. The default is **oracle/pdc**. • **imageTag**: The tag name for the PDC image. The default is **:16.26.0.0.0**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
deployment.fmw.*	**ocpdc**	The details for pulling WebLogic images for PDC from a container registry: • **imageRepository**: The name of the container registry from which to pull the WebLogic image. The default is **container-registry.oracle.com/**. • **imageName**: The name of the container repository from which to pull the WebLogic image. The default is **middleware/fmw-infrastructure**. • **imageTag**: The tag name for the WebLogic image. The default is **:14.1.2.0-jdk21-ol9**. See "Pulling WebLogic Images for PDC, Billing Care, Billing Care REST API, and Business Operations Center" for more information.
nodeSelector	**ocpdc**	The name of the node on which to run the following PDC pods: • pdc • RRE • BRE • SyncPDC • Import-Export Set this key if you want to constrain the PDC pods to run only on the node you specify. For more information, see "nodeSelector" in the Kubernetes documentation. **Note**: To override the rules for a specific PDC pod, specify a different value for the pod's **nodeSelector** key. For example, set the **ocpdc.configEnv.transformation.nodeSelector** key to apply rules specifically to the Transformation pod.
affinity	**ocpdc**	The rules for running the following PDC pods on specific nodes: • pdc • RRE • BRE • SyncPDC • Import-Export Set this key if you want to constrain the PDC pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation. **Note**: To override the rules for a specific PDC pod, specify a different value for the pod's **affinity** key. For example, set the **ocpdc.configEnv.transformation.affinity** key to apply rules specifically to the Transformation pod.
addOnPodSpec	**ocpdc**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.
domainUID	**ocpdc.wop**	The name of the PDC WebLogic Server domain. The default is **pdc-domain**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
resources.*	**ocpdc**	The minimum and maximum CPU and memory resources for the PDC domain admin-server pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
configEnv.*	**ocpdc**	The configuration details for PDC. • **name**: The name of the PDC ConfigMap. The default is **pdc-configmap-env**. • **upgrade**: Set to **true** to upgrade PDC. • **dbHostName**: The host name of the PDC and cross-reference database. The value must match that of **oc-cn-op-job-helm-chart**. • **dbPort**: The port for the PDC and cross-reference database. The value must match that of **oc-cn-op-job-helm-chart**. • **dbService**: The service name for the PDC and cross-reference database. The value must match that of **oc-cn-op-job-helm-chart**. • **dbSSLMode**: The type of SSL connection required for connecting to the cross-reference database: two-way SSL authentication is required (**TWO_WAY**), one-way SSL authentication is required (**ONE_WAY**), or SSL authentication is not required (**NO**). The default is **NO**.
extPDCDBSSLWalletSecret	**ocpdc.configEnv**	The names of the external Kubernetes Secret containing the custom SSL database wallet for PDC. This value must match the one for the **oc-cn-op-job-helm-chart**. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
transformation.*	**ocpdc.configEnv**	The details for the transformation. • **logLevel**: The logging level, which can be **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, or **FINEST**. The default is **WARNING**. • **logFileSize**: The maximum file size, in bytes, of the log files. After the log file meets the maximum, PDC closes the log file and creates a new log file. The default is **500000**. See "Rotating PDC Log Files" in *BRM Cloud Native System Administrator's Guide* for more information. • **logFileCount**: The maximum number of log files to retain for the application. The default is **10**. • **persistTransactionLogs**: Whether to persist log files in the database after they are closed. Possible values are **all**, **disabled**, and **failed**. The default is **failed**, which specifies to persist only failed transaction logs. • **MEM_ARGS**: The memory argument, surrounded by quotes. For example: **"-Xms1024m -Xmx2048m -XX:CompileThreshold=8000"**. • **persistOutFiles**: Whether to keep the output and payload files generated by the transformation engine in the **out** directory (**enabled**) or remove the files from the out directory (**disabled**). The default is **disabled**. • **resources.***: The minimum and maximum CPU and memory resources for the RRE and BRE pods. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The name of the node on which to run the RRE and BRE pods. • **affinity**: The rules for running the RRE and BRE pods on specific nodes.
seedData.*	**ocpdc.configEnv**	The details for deploying seed data. **Note**: If balance element and RUM data already exist in the PDC database, they are not overwritten. • **BE**: Whether to load sample balance elements into the PDC database when PDC is deployed (**true**) or not (**false**). The default is **false**. **Note**: If balance element data already exists in the PDC database, it is not overwritten. • **RUM**: Whether to load sample RUMs into the PDC database when PDC is deployed (**true**) or not (**false**). The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
importExport.*	**ocpdc.configEnv**	The default for running the **ImportExportPricing** utility in PDC cloud native. For more information, see "Running PDC Applications" in *BRM Cloud Native System Administrator's Guide*. • **IE_Operation.***: The operation for the **ImportExportPricing** utility to perform. When you deploy PDC, ensure that this key has an empty value. – Empty value: No operation is performed. This is the default. – **export**: The utility exports data from the PDC database into an XML file. – **import**: The utility imports data from the XML file into the PDC database. – **publish**: The utility publishes components from PDC to the batch rating engine, real-time rating engine, or ECE. – **changeSetStatus**: Checks the current status of one or more changesets in PDC. – **keep**: The utility retains the latest version of successfully promoted PDC components. – **deleteprofile**: The utility deletes pricing profiles from PDC. – **type**: The utility displays the pricing or setup components available in PDC. • **IE_Component.***: The type of components and objects to import or export using the **ImportExportPricing** utility. Don't include the hyphen (**-**) prefix with the value. – **config**: Imports or exports setup components, such as tax codes, business profiles, and general ledger IDs. – **pricing**: Imports or exports pricing components, such as events, charges, and chargeshares. – **metadata**: Imports or exports the event, service, account, and profile attribute specifications. – **profile**: Imports or exports pricing profile data. – **customfields**: Imports or exports custom fields. – **brmobject**: Exports BRM configuration objects such as services, events, and G/L IDs. – **all**: Imports or exports all objects and components. • **IE_File_OR_Dir_Name**: The name of the XML file or **ImportExport** directory that contains the list of components and objects to import into the PDC database. This XML file is used by the **ImportExportPricing** utility. If importing or deleting components, copy the XML file to the HostPath specified in **pdcBRMHostPath** or to pdc-brm-pvc. Set the appropriate read and write permissions for files or directories, and set the ownership to *runAsUser* using **chown** *runAsUser*:**0**. • **extraCmdLineArgs**: The extra command-line arguments for the **ImportExportPricing** utility, apart from operation,

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
		component, and file name. The value must be surrounded by quotes. For example, "**-n** *ObjectName***"**. A new **separator** option is now supported for the export operation. It specifies the character used to split multiple pricing or configuration object names. Place this option before the **name** parameter. If not specified, the utility uses a comma (,) as the default separator. Use this option when any object name contains a comma. • **logLevel**: Sets the logging level, which can be **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, or **FINEST**. The default is **WARNING**. • **logSize**: Sets the maximum file size, in bytes, of the log files. After the log file meets the maximum, PDC closes the log file and creates a new log file. • **logCount**: Specifies the maximum number of log files to retain for the application. See "Rotating PDC Log Files" in *BRM Cloud Native System Administrator's Guide* for more information. • **persistIELogs**: Specifies whether to persist log files in the database after they are closed. Possible values are **all**, **disabled**, and **failed**. The default is **failed**. • **xmlStatusFilesToKeep:** Specifies the maximum number of XML status files to retain. They are stored in the **ImportExportPricing** log location. The default is **10**. If the value is 0, the utility does not generate new XML status files, but setting the value to **0** does not delete existing files. If the value is a positive integer, the utility retains the specified number of latest XML status files and deletes older files. • For **changeSetStatus**, set **extraCmdLineArgs** to one of the following values: – "**-name** *changesetName***"**: Specifies one or more changeset names. To specify multiple changesets, separate the names with commas. – "**-id** *changesetId***"**: Specifies one or more changeset IDs. To specify multiple changesets, separate the IDs with commas. • **resources.***: The minimum and maximum CPU and memory resources for the Import-Export pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the Import-Export pod on specific nodes. • **affinity**: The rules for deploying the Import-Export pod on specific nodes.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
syncPDC.*	**ocpdc.configEnv**	The details for running the **SyncPDC** utility in PDC cloud native. For more information, see "Running PDC Applications" in *BRM Cloud Native System Administrator's Guide*. • **upgradeFromPS2**: Set this to true to upgrade from 12.0 Patch Set 2 to 12.0 Patch Set 4 or higher. • **logLevel**: Sets the logging level, which can be **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, or **FINEST**. The default is **WARNING**. • **logFileSize**: Sets the maximum file size, in bytes, of the log files. After the log file meets the maximum, PDC closes the log file and creates a new log file. The default is **20000**. • **logFileCount**: Specifies the maximum number of log files to retain for the application. The default is **10**. See "Rotating PDC Log Files" in *BRM Cloud Native System Administrator's Guide* for more information. • **MEM_ARGS**: The memory argument. • **skipBREMigration**: Skips the synchronization of pipeline configuration data. The default is **false**. This key is ignored when ECE is enabled in a PDC system. • **syncPDCStartAt**: The scheduled time for running the **SyncPDC** utility. This key is set at deployment time only. Valid values include: **startAt** and **"*HH*:*MM*"**. The default is **23:59**. • **syncPDCInterval**: The scheduled frequency for running the **SyncPDC** utility. This key is set at deployment time only. The default is **1:D**. • **enrichmentFileName**: Set this to **ECEEventEnrichmentSpec.xml**. Store the enrichment file in the path specified in **pdcBrmHostPath**. This is applicable at both PDC deployment time and individual **SyncPDC** runtime. • **runSyncPDC**: Whether to create the SyncPDC pod and start the BRM-to-PDC synchronization process (**true**) or delete the SyncPDC pod and stop the synchronization process (**false**). The default is **true**. • **resources.***: The minimum and maximum CPU and memory resources for the SyncPDC pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector**: The rules for deploying the SyncPDC pod on specific nodes. • **affinity**: The rules for deploying the SyncPDC pod on specific nodes.
monitoring.*	**ocpdc.configEnv**	The details for monitoring PDC and collecting WebLogic metrics. By default, monitoring is disabled. For more information, see "Monitoring PDC in a Cloud Native Environment" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-6 (Cont.) Pricing Design Center Keys for oc-cn-helm-chart

Key	Path in values.yaml	Description
secretValue.*	**ocpdc**	The passwords for PDC. • **name**: The name of the Secret key. The default is **pdc-secret-env**. • **walletPassword**: The passwords for the PDC application wallet and PDC BRM integration wallet.
secret.*	**ocpdc**	The details for the external Kubernetes Secret: • **extPdcBrmIntegrationSecret**: The name of the external Kubernetes Secret for the PDC-BRM integration. • **extPdcSecret**: The name of the external Kubernetes Secret for the PDC job. • **extPdcFmwSecret**: The name of the external Kubernetes Secret for PDC Fusion Middleware. • **extPdcFmwRcuSecret**: The name of the external Kubernetes Secret for the PDC Fusion Middleware RCU schema. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Adding PDC Keys for oc-cn-op-job-helm-chart

You must create an **override-values.yaml** for **oc-cn-op-job-helm-chart** and then add the PDC-specific keys in Table 7-7.

For information about all PDC-specific keys, see the descriptions in the **oc-cn-op-job-helm-chart/values.yaml** file.

> ⚠ **Caution**
>
> Keys with the path **ocpdc.secretValue** hold sensitive data. Handle them carefully with controlled access to the file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Table 7-7 Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
isEnabled	**ocpdc**	Whether to enable PDC jobs: • **true**: Enables PDC jobs. This is the default. • **false**: Disables PDC jobs.
isClean	**ocpdc**	Whether to clean old PDC deployment and instance logs: • **true**: Removes all existing PDC deployment and instance logs. This is the default. • **false**: Keeps all existing PDC logs.
lang	**ocpdc**	The Linux system locale. The default is **en_US.UTF-8**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
tz	**ocpdc**	The Linux time zone. The default is **UTC**.
volume.domain.createOption.*	**ocpdc**	By default, PDC uses dynamic volume provisioning. PDC on WebLogic Kubernetes Operator requires shared storage for the WebLogic domain home and configuration in clustered deployments. Use a storage class that supports ReadWriteMany, or an equivalent shared storage solution such as NFS. To use a static volume instead, add the **createOption** keys under the **volume** section. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. **Note**: Ensure you provide the required permissions to the volume path by following the guidelines in "Persistent Volume Storage Locations" in the WebLogic Kubernetes Operator documentation.
storageSize	**ocpdc**	The storage size for the pdc-brm-pv or pdc-brm-pvc, such as 10 Gi. The storage must use an access mode supported for PDC shared storage in your deployment.
deployment.*	**ocpdc**	The details for deploying the PDC image: • **imageName**: The name of the PDC image. The default is **oracle/pdc**. • **imageTag**: The tag name for the PDC image. The default is **:16.26.0.0.0**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.
deployment.fmw.*	**ocpdc**	The details for pulling WebLogic images for PDC from a container registry: • **imageRepository**: The name of the container registry from which to pull the WebLogic image. The default is **container-registry.oracle.com/**. • **imageName**: The name of the container repository from which to pull the WebLogic image. The default is **middleware/fmw-infrastructure**. • **imageTag**: The tag name for the WebLogic image. The default is **:14.1.2.0-jdk21-ol9**. See "Pulling WebLogic Images for PDC, Billing Care, Billing Care REST API, and Business Operations Center" for more information.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
nodeSelector	**ocpdc**	The rules for scheduling a PDC job pod on a particular node using **nodeSelector** or **affinity**. **pdc-domain-job**: Set this key to constrain the PDC pods to run only on the node you specify. For more information, see "nodeSelector" in the Kubernetes documentation. **Note**: To override the rules for a specific PDC pod, specify a different value for the pod's **nodeSelector** key. For example, set the **ocpdc.configEnv.transformation.nodeSelector** key to apply rules specifically to the Transformation pod.
affinity	**ocpdc**	The rules for scheduling a PDC job pod on a particular node using **nodeSelector** or **affinity**. **pdc-domain-job**: Set this key if you want to constrain the PDC pod to run only on nodes that meet your criteria. For more information, see "Node Affinity" in the Kubernetes documentation. **Note**: To override the rules for a specific PDC pod, specify a different value for the pod's affinity key. For example, set the **ocpdc.configEnv.transformation.affinity** key to apply rules specifically to the Transformation pod.
addOnPodSpec	**ocpdc**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
wop.*	**ocpdc**	The details for deploying the WebLogic Operator. • **domainUID**: The name of this PDC WebLogic Server domain. • **includeServerOutInPodLog**: Whether to include the server out file in the pod's stdout log (**true**) or not (**false**). The default is **true**. • **jtaTimeoutSeconds**: The maximum amount of time, in seconds, an active transaction is allowed to be in the first phase of a two-phase commit transaction. The default is **10000**. If the time expires, the transaction is automatically rolled back. • **jtaAbandonTimeoutSeconds**: The maximum amount of time, in seconds, a transaction manager continues to attempt completing the second phase of a two-phase commit transaction. The default is **10000**. • **stuckThreadMaxTime**: The number of seconds a thread must be continually working before the server considers the thread to be stuck. The default is **20000**. • **idlePeriodsUntilTimeout**: The number of idle periods until the peer is considered to be unreachable. The default is **40**. • **dataSourceXaTxnTimeout**: The number of seconds until the data source transaction times out. The default is **0**. When set to **0**, the WebLogic Server Transaction Manager passes the global WebLogic Server transaction timeout in seconds in the method. • **pdcAppSesTimeOut**: The PDC application (**pricingui.ear**) session time out, in seconds. The default is **36000**. • **pdcAppSesInvInterTimeOut**: The PDC application (**pricingui.ear**) session invalid interval time out, in seconds. The default is **3000**. • **maxMessageSize**: The maximum number of bytes allowed in messages that are received over supported protocols. The default is **10000000**. • **users.***: The users to add to the PDC domain. For more information, see "Creating PDC Users" in *BRM Cloud Native System Administrator's Guide*.
resources.*	**ocpdc**	The minimum and maximum CPU and memory resources for the PDC domain job. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
PDC Ports	**ocpdc**	The details about the PDC ports to use. • **exposePorts**: Exposes the SSL HTTPS port only (**yes**), the HTTP port only (**no**), or both (**all**). The default is **all**. • **t3ChannelPort**: The port number for the t3 channel. The default is **30799**. Use this key if PDC needs to use the t3 protocol to communicate with an external system, such as Elastic Charging Engine (ECE). Set this to a Kubernetes port number from 30000 through 32767 that is not in use. • **t3ChannelAddress**: The IP address for the primary node or load balancer. • **t3sChannelPort**: The port number for the t3s channel. The default is **30800**. Use this key if PDC needs to use the t3s protocol to communicate with an external system such as ECE. Set this to a Kubernetes port number from 30000 through 32767 that is not in use. • **t3sChannelAddress**: If SSL is enabled in the WebLogic domain, set this to the IP address for the primary node or load balancer.
Java Settings	**ocpdc.configEnv**	The Java security settings. • **USER_MEM_ARGS**: The custom memory arguments for WebLogic Admin Server. • **USER_JAVA_OPTIONS**: The custom Java options for WebLogic Admin Server. • **javaSecurityFileName**: The name of the Java security file listing the ciphers to disable. Place the file in the **pdc/java_security** directory.
tlsVersions	**ocpdc.configEnv**	The list of TLS versions to support for connection with the WebLogic domain. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
Log File Rotation	**ocpdc.configEnv**	The settings for rotating Pricing Server log and tracing log files: • **pdcAppLogLevel**: Sets the logging level, which can be **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, or **FINEST**. The default is **WARNING**. • **pdcAppLogFileSize**: Sets the maximum file size, in bytes, of the log files. After the log file meets the maximum, PDC closes the log file and creates a new log file. • **pdcAppLogFileCount**: Specifies the maximum number of log files to retain for the application. See "Rotating PDC Log Files" in *BRM Cloud Native System Administrator's Guide* for more information.
RCU Schema	**ocpdc.configEnv**	The details about the RCU schema. • **rcuJdbcURL**: The connection string for connecting to a database where schemas needed by Oracle Fusion Middleware products will be created, especially OPSS. Use the format "*host***:***port***/***service*". • **rcuPrefix**: The prefix for the PDC domain RCU schema. For example, if the prefix is **XYZ** and the schema name is **STB**, the PDC domain RCU schema name will be **XYZ_STB**. • **rcuRecreate**: Whether to re-create the PDC domain if present (**true**) or not (**false**).
isCustomWLSPython	**ocpdc.configEnv**	Whether to run your custom WebLogic Python files: • **true**: Run your custom WebLogic Python files located in **oc-cn-op-job-chart/pdc/customWLSPython**. • **false**: Do not run a custom WebLogic Python file. Set the appropriate read and write permissions for each file, and set the ownership to **1000** using **chown 1000:0**.
addOPSSWallet	**ocpdc.configEnv**	Whether you added the OPSS wallet to the Helm chart. • **true**: The OPSS wallet is added to the Helm chart. If set to **true**, you must copy the OPSS wallet file from **pdc-app-pvc/stores/opss_wallet/** to **oc-cn-op-job-helm-chart/pdc/opss_wallet**. • **false**: The OPSS wallet was not added to the Helm chart. This is the default. **Note**: The OPSS wallet file will not be available in **pdc-app-pvc/stores/opss_wallet** the first time you run **oc-cn-op-job-helm-chart** or create a new RCU prefix.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
extOPSSWallet	**ocpdc.configEnv**	The name of the external Kubernetes Secret containing the custom OPSS wallet file for PDC. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
honorOMF	**ocpdc.configEnv**	Whether the RDS database honors the Oracle-Managed Files (OMF) naming format: • **true**: OMF format is used. • **false**: OMF format is not used. This is the default.
keyStoreType	**ocpdc.configEnv**	The SSL KeyStore type for the PDC domain. The default is **JKS**.
keyStoreAlias	**ocpdc.configEnv**	The alias name for the PDC domain SSL KeyStore.
keyStoreIdentityFileName	**ocpdc.configEnv**	The name of the PDC domain SSL identity keystore file. Before deploying PDC, generate the JKS identity keystore and place it in **oc-cn-op-job-helm-chart/pdc/pdc_keystore**. PDC does not ship or create any certificates during deployment. Customers must generate and/or place custom certificates before PDC deployment. For information, see "Configuring SSL for the PDC WebLogic Domain".
keyStoreTrustFileName	**ocpdc.configEnv**	The name of the PDC domain SSL trust keystore file. Before deploying PDC, generate the JKS trust keystore and place it in **oc-cn-op-job-helm-chart/pdc/pdc_keystore**. PDC does not ship or create any certificates during deployment. Customers must generate and/or place custom certificates before PDC deployment. For information, see "Configuring SSL for the PDC WebLogic Domain". .
extPDCKeystoreSecret	**ocpdc.configEnv**	The name of the external Kubernetes Secret containing the custom KeyStore certificate file for PDC. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
isSSOEnabled	**ocpdc.configEnv**	Set to **true** to configure and use SAML 2.0 SSO service. The default is **false**.
extMetadataCM	**ocpdc.configEnv**	The name of the external ConfigMap containing the IDP metadata file.
samlAsserterName	**ocpdc.configEnv**	The name of the SAML Asserter. It should be the same as OEM or IDCS. The default is **pdcSAML2IdentityAsserter**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
ssoPublishedSiteURL	**ocpdc.configEnv**	The base URL used to construct endpoint URLs, typically, the load balancer host and port at which the server is visible externally. It must be appended with **/saml2**. For example: **https://** *LoadBalancerHost*:*LoadBalancerPort*/**saml2**.
ssoDefaultURL	**ocpdc.configEnv**	The URL to which unsolicited authentication responses are sent if they do not contain an accompanying target URL.
ssoLogoutURL	**ocpdc.configEnv**	The URL where users are redirected after they log out from the application (OEM or IDCS log out).
Database Details	**ocpdc.configEnv**	The details for • **dbHostName**: The host name of the PDC and cross-reference database. • **dbPort**: The port for the PDC and cross-reference database. • **dbService**: The service name for the PDC and cross-reference database. • **dbSysDBAUser**: The SYS, System, or Sys DBA user for the PDC and cross-reference database. If this key is not configured, PDC assumes that **pdcSchemaUserName** and **crossRefSchemaUserName** are already present in the database with the required permissions. • **dbSysDBARole**: The role of the PDC and cross-reference database SYS, System, or Sys DBA user. • **dbSSLMode**: The type of SSL connection required for connecting to the database: **TWO_WAY**, **ONE_WAY**, or **NO**. • **dbWalletType**: The type of file specified as the TrustStore for SSL connections: **SSO** or **pkcs12**. The default is **SSO**. If set to **ONE_WAY** or **TWO_WAY**, place the database wallet in the **oc-cn-helm-chart/pdc/pdc_db_wallet** directory. Create the directory structure if it is not present and do not change the directory name. • **extPDCDBSSLWalletSecret**: The name of the Kubernetes Secret containing the custom SSL database wallet file for PDC. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
Database Schema	**ocpdc.configEnv**	The details about the PDC database schema. • **crossRefSchemaPDCTableSpace**: The name of the PDC tablespace for the transformation cross-reference schema. This field is case-sensitive. • **crossRefSchemaTempTableSpace**: The name of the temporary tablespace for the transformation cross-reference schema. This field is case-sensitive. • **crossRefSchemaUserName**: The cross-reference database schema user name. • **pdcSchemaPDCTableSpace**: The tablespace name of the PDC schema. This field is case-sensitive. • **pdcSchemaTempTableSpace**: The tablespace name of the temporary schema. This field is case-sensitive. • **pdcSchemaUserName**: The PDC database schema user name. • **rcuWalletSchemaUserName**: The RCU wallet schema user name. The default schema user name is **PDCRCUWALLET**. **Note:** The OPSS wallet file created for the RCU schema is stored in the RCU_WALLET_DETAILS table during the first run. If the wallet file is available for the given RCU prefix, it is reused in subsequent runs and the RCU schema is not re-created. If the OPSS wallet file is present in **oc-cn-op-job-helm-chart/pdc/opss_wallet**, it takes precedence.
pdcAdminUser	**ocpdc.configEnv**	The PDC admin user name, which includes the Pricing Design Admin role. The default is **cnepdcadminuser**.
supportECE	**ocpdc.configEnv**	The charging engine to use: Elastic Charging Engine (**true**) or the real-time and batch rating engine (**false**). The default is **true**.
Upgrade	**ocpdc.configEnv**	The details for upgrading PDC. • **siteNo**: Set to **2** for zero-downtime upgrades (ZDU). The default is **1**. • **upgrade**: Set to **true** to upgrade from a previous version to a 15.*x* Patch Set or to deploy a 15.*x* interim patch. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-7 (Cont.) Pricing Design Center Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml	Description
secretValue.*	**ocpdc**	The credentials for accessing the system. • **adminPassword**: The password for the WebLogic domain's administrative user, which is used for accessing the WebLogic Console for administrative operations. • **rcuSchemaPassword**: The password for the Oracle Fusion Middleware product schemas that will be created by RCU and used by OPSS. • **keyStoreIdentityKeyPass**: The password for the PDC domain SSL Identity key. • **keyStoreIdentityStorePass**: The password for the PDC domain SSL identity store. • **keyStoreTrustStorePass**: The password for the PDC domain SSL TrustStore. • **dbPassword**: The Sys or System user password for the PDC and Cross Reference schema. • **pdcSchemaPassword**: The password for the PDC database schema user. • **crossRefSchemaPassword**: The password for the transformation cross-reference database schema user. • **rcuWalletSchemaPassword**: The password for the PDC RCU OPSS wallet schema. • **dbWalletPassword**: The password for the database SSL wallet. This key is required if **dbWalletType** is set to **pkcs12**. • **walletPassword**: The passwords for the PDC application wallet and PDC BRM integration wallet. • **pdcAdminUserPassword**: The password for the PDC admin user, which includes the Pricing Design Admin role.
service.*	**ocpdc**	The pdc-service service's details. • **name**: The name of the service: **pdc-service**. • **type**: The service type. The default is **ClusterIP**.

Configuring SSL for the PDC WebLogic Domain

To configure SSL for the PDC WebLogic domain:

1. In oc-cn-op-job-helm-chart, set **ocpdc.configEnv.exposePorts** to one of the following:
   - **yes** if you use HTTPS only
   - **no** if you use HTTP only
   - **all** if you use both
2. Generate the identity and trust JKS keystores.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3. Copy the identity keystore, trust keystore, and any required public certificates to the PDC keystore directory: **oc-cn-op-job-helm-chart/pdc/pdc_keystore**.
4. In the **oc-cn-op-job-helm-chart** override file, configure the following:

```
ocpdc:
  configEnv:
    exposePorts: "exposePorts"
    keyStoreType: "keystoreType"
    keyStoreAlias: "CustomIdentityAlias"
    keyStoreIdentityFileName: "identityKeystoreFileName"
    keyStoreTrustFileName: "trustKeystoreFileName"
    extPDCKeystoreSecret: "customKeystoreSecretName"
  secretValue:
    keyStoreIdentityKeyPass: "identityKeyPassword"
    keyStoreIdentityStorePass: "identityStorePassword"
    keyStoreTrustStorePass: "trustStorePassword"
```

where:

- *exposePorts* specifies whether to expose ports: **yes**, **no**, or **all**.
- *keystoreType* is the keystore format or type.
- *CustomIdentityAlias* is the PDC domain alias in the identity keystore.
- *identityKeystoreFileName* is the identity keystore file name.
- *trustKeystoreFileName* is the trust keystore file name.
- (Optional) *customKeystoreSecretName* is a custom Kubernetes secret containing PDC keystore material. If **ocpdc.configEnv.extPDCKeystoreSecret** is set, PDC uses the keystore material from the external Kubernetes Secret and does not rely on local files in the **oc-cn-op-job-helm-chart/pdc/pdc_keystore** directory.
- *identityKeyPassword*, *identityStorePassword*, and *trustStorePassword* are managed-secret values for the respective keystore passwords. You can provide them using either chart-managed secret values or an external Kubernetes Secret.

> **Note**
>
> Provide the password values through the approved deployment secret process. Do not commit plain-text passwords to the **values.yaml** file or any **override-values.yaml** file.

5. Deploy or redeploy PDC. Validate that the PDC HTTPS endpoint is reachable and has no TLS or keystore errors.

Setting Up SSO for PDC Cloud Native

SSO allows users to log in to applications using a single user name and password combination. You set up SSO for PDC cloud native services by using SAML 2.0.

To set up SSO for PDC:

1. Export the SAML 2.0 metadata XML file from your identity and access management (IAM) system.

Copyright © 2019, 2026, Oracle and/or its affiliates.

For example, if you are using Oracle Access Management, you can export the file by following the instructions in "Exporting Metadata" in *Oracle Fusion Middleware Administering Oracle Access Management*.

2. Add the metadata XML file to your BRM cloud native deployment by doing one of the following:
   - Rename the metadata XML file to **metadata.xml**, and then move **metadata.xml** to the **oc-cn-op-job-helm-chart/pdc/idp** directory.
   - Pre-create the IDP metadata ConfigMap for PDC and set the **ocpdc.configEnv.extMetadataCM** key in your **override-values.yaml** file for **oc-cn-op-job-helm-chart**.
3. Configure the KeyStores needed by SAML by doing one of the following:
   - Generate the Identity and Trust KeyStores and then move your files under the **oc-cn-op-job-helm-chart/pdc/pdc_keystore** directory.
   - Pre-create the Kubernetes Secret for the Identity and Trust KeyStore files and set the **ocpdc.configEnv.extKeystoreSecret** key in your **override-values.yaml** file for both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**.

     For more information, see "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
4. In your **override-values.yaml** file for **oc-cn-op-job-helm-chart**, set the following keys:
   - **ocpdc.configEnv.isSSOEnabled**: Set this to **true**.
   - **ocpdc.configEnv.keyStoreAlias**: Set this to the private key alias of the KeyStore.
   - **ocpdc.configEnv.keyStoreType**: Set this to the file type of the SSL Identity and Trust KeyStore. The only supported value is **JKS**.
   - **ocpdc.configEnv.keyStoreIdentityFileName**: Set this to the name of the Identity KeyStore file.
   - **ocpdc.configEnv.keyStoreTrustFileName**: Set this to the name of the Trust KeyStore file.
   - **ocpdc.configEnv.samlAsserterName**: Set this to the name of the SAML Asserter. The default is **pdcSAML2IdentityAsserter**.
   - **ocpdc.configEnv.ssoPublishedSiteURL**: Set this to the base URL used to construct endpoint URLs. This is typically the load balancer host and port where the server is visible externally. It must be appended with **/saml2**. For example: **https://** *LoadBalancerHost*:*LoadBalancerPort*/**saml2**.
   - **ocpdc.configEnv.ssoDefaultURL**: Set this to the URL where unsolicited authentication responses are sent if they do not contain an accompanying target URL.
   - **ocpdc.secretValue.keyStoreIdentityStorePass**: Set this to the StorePass for the Identity KeyStore.
   - **ocpdc.secretValue.keyStoreIdentityKeyPass**: Set this to the KeyPass for the Identity KeyStore.
   - **ocpdc.secretValue.keyStoreTrustStorePass**: Set this to the StorePass for the Trust KeyStore.
5. Configure your load balancer's rules to send responses to the PDC WebLogic domain with **/saml2** appended to the URL path.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> Add this rule to your existing load balancer rules for routing responses to PDC (***/pdc***), the load balancer host name, and so on.

See "Exposing BRM Services Outside the Kubernetes Cluster".

6. Deploy your PDC cloud native services by following the instructions in "Deploying BRM Cloud Native Services".
7. After PDC is deployed, retrieve the **sp-metadata-admin-server.xml** file from the ***/shared/applications/****PDC_HOME*/*domainUID* directory in your container, where *domainUID* is the name of your PDC domain specified in the **ocpdc.wop.domainUID** key.

   The XML file configures the Web SSO Provider Partner. It contains the partner's KeyStore certificates, SAML assertion details, and the URLs where the SAML Identity Provider redirects to provide access to PDC.
8. Create a profile for your identity provider partner by loading the **sp-metadata-admin-server.xml** file into your IAM system.

   For example, if you are using Oracle Access Management, you can load the file by following the instructions in "Creating Remote Identity Provider Partners" in *Oracle Fusion Middleware Administering Oracle Access Management*.

Configuring Pipeline Configuration Center

To configure Pipeline Configuration Center (PCC) to run in your BRM cloud native environment:

1. Override the PCC-specific keys in the **values.yaml** file for **oc-cn-op-job-helm-chart**. See "Adding Pipeline Configuration Center Keys for oc-cn-op-job-helm-chart".
2. Override the PCC-specific keys in the **values.yaml** file for **oc-cn-helm-chart**. See "Adding Pipeline Configuration Center Keys for oc-cn-helm-chart".
3. Set up volume mounts. See "About PCC Volume Mounts".
4. Create a WebLogic domain and install the PCC application. See "Creating a WebLogic Domain and Installing the PCC Application".
5. Set up SAML for SSO in PCC. See "Setting Up SSO for Pipeline Configuration Center".
6. Set up local users and groups for PCC. See "Setting Up Local Users and Groups for PCC".
7. Start and stop your WebLogic servers. See "Starting and Stopping WebLogic Servers".
8. Enable SSL in PCC. See "Configuring SSL in PCC".

Adding Pipeline Configuration Center Keys for oc-cn-op-job-helm-chart

Table 7-8 lists the keys that directly impact PCC deployment. Add these keys to your **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-8 PCC Keys for oc-cn-op-job-helm-chart

Key	Paths in values.yaml File	Description
isEnabled	**ocpcc.pcc**	Whether to deploy, configure, and start PCC services: • **false**: Does not create the Kubernetes resources for using PCC. • **true**: Creates the Kubernetes resources for using PCC. This is the default.
imageName	**ocpcc.pcc.deployment.app**	The name of the PCC image, such as **oracle/pcc**
imageTag	**ocpcc.pcc.deployment.app**	The tag associated with the image. This is generally the patch set number prefixed with a colon (**:**). For example, :**16.26.0.0.0**
dbSSLMode	**ocpcc.pcc.configEnv**	The type of connection required to connect to the database: • **TWO_WAY**: Two-way SSL authentication is required. In this case, both the client and server must authenticate each other's identity. • **ONE_WAY**: One-way SSL authentication is required. In this case, the client must authenticate the server's identity. This is the default. • **NO**: SSL authentication is not required.
dbWalletType	**ocpcc.pcc.configEnv**	The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO**or **PKCS12**.
rcuJdbcURL	**ocpcc.pcc.configEnv**	The connection string for connecting to the database where schemas needed by Oracle Fusion Middleware products will be created, especially OPSS.
rcuDBARole	**ocpcc.pcc.configEnv**	The role of the database administrator user.
rcuArgs	**ocpcc.pcc.configEnv**	The additional arguments for creating the RCU.
ldapHost	**ocpcc.pcc.configEnv**	The host name or IP address of the LDAP Server (for example, OUD) where users and groups will be configured for access to PCC.
ldapPort	**ocpcc.pcc.configEnv**	The port number on which the LDAP server is listening.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-8 (Cont.) PCC Keys for oc-cn-op-job-helm-chart

Key	Paths in values.yaml File	Description
ldapGroupBase	**ocpcc.pcc.configEnv**	The LDAP base DN that contains groups.
ldapUserBase	**ocpcc.pcc.configEnv**	The LDAP base DN that contains users.
extDBSSLWalletSecret **extKeystoreSecret**	**ocpcc.pcc.configEnv**	The names of the pre-created Kubernetes Secrets for the Pipeline Configuration Center KeyStore certificates and wallets. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
keystoreAlias	**ocpcc.pcc.configEnv**	The private key alias of the KeyStore.
keystoreType	**ocpcc.pcc.configEnv**	The file type of the SSL Identity and TrustStore, which is either **PKCS12**or **JKS**. The default is **PKCS12**.
keystoreIdentityFileName	**ocpcc.pcc.configEnv**	The file name of the Identity KeyStore.
keystoreTrustFileName	**ocpcc.pcc.configEnv**	The file name of the Trust KeyStore.
isSSOEnabled	**ocpcc.pcc.configEnv**	Whether to enable single sign-on (SSO) for PCC cloud native services through SAML 2.0: • **true**: **SSO** is enabled for PCC cloud native services. • **false**: **SSO** is disabled. This is the default.
samlAsserterName	**ocpcc.pcc.configEnv**	The name of the SAML Asserter. The default is **samlPCCAsserter**.
ssoPublishedSiteURL	**ocpcc.pcc.configEnv**	The base URL that is used to construct endpoint URLs. This is typically the Load Balancer host and port at which the server is visible externally. It must be appended with **/saml2**. For example: **https://***LoadBalancerHost***:***LoadBalancerPort***/saml2**.
ssoDefaultURL	**ocpcc.pcc.configEnv**	The URL where unsolicited authentication responses are sent if they do not contain an accompanying target URL.
reloadVersion	**ocpcc.pcc.configEnv**	Update this value with any value different from the current value to force a restart of the deployer.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-8 (Cont.) PCC Keys for oc-cn-op-job-helm-chart

Key	Paths in values.yaml File	Description
adminPassword	**ocpcc.pcc.secretVal**	The password of the WebLogic domain's administrative user, which is used for accessing the WebLogic Console for administrative operations.
ldapPassword	**ocpcc.pcc.secretVal**	The password of the LDAP Server admin user.
rcuSysDBAPassword	**ocpcc.pcc.secretVal**	The password for the rcuJdbcURL database administrator.
rcuSchemaPassword	**ocpcc.pcc.secretVal**	The passwords for the schemas of Oracle Fusion Middleware products that will be created by RCU, which is used by OPSS.
dbWalletPassword	**ocpcc.pcc.secretVal**	The password for accessing the certificates from the TrustStore and KeyStore.
keystoreIdentityPassword	**ocpcc.pcc.secretVal**	The StorePass for the Identity KeyStore.
keystoreKeyPassword	**ocpcc.pcc.secretVal**	The KeyPass for the Identity KeyStore.
keystoreTrustPassword	**ocpcc.pcc.secretVal**	The StorePass for the Trust KeyStore.
domainUID	**ocpcc.pcc.wop**	The name of the domain. The default is **pcc-domain**.
adminChannelPort	**ocpcc.pcc.wop**	The NodePort where the admin-server's HTTP service is accessible.
serverStartPolicy	**ocpcc.pcc.wop**	The WebLogic servers that the Operator starts when it discovers the domain: • **NEVER**: Does not start any server in the domain. • **ADMIN_ONLY**: Starts only the administration server (no managed servers will be started). • **IF_NEEDED**: Starts the administration server and clustered servers up to the replica count.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-8 (Cont.) PCC Keys for oc-cn-op-job-helm-chart

Key	Paths in values.yaml File	Description
volume.*	**ocpcc.pcc**	Details about the PVC for the pcc pod: • **storage**: The storage size of the volume. • **createOption**: By default, the pcc pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.
nodeSelector	**ocpcc.pcc**	The node selector rules for scheduling WebLogic Server pods on particular nodes using simple selectors.
affinity	**ocpcc.pcc**	The affinity rules for scheduling WebLogic Server pods on particular nodes using more powerful selectors.

Adding Pipeline Configuration Center Keys for oc-cn-helm-chart

Table 7-9 lists the keys that directly impact PCC deployment. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart**.

Table 7-9 Pipeline Configuration Center Keys

Key	Path in Values.yaml File	Description
appLogLevel	**ocpcc**	The logging level at which application logs must be captured in log files: **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, **FINEST**, and **ALL**.
isEnabled	**ocpcc.pcc**	Whether to deploy, configure, and start PCC services: • **false**: Does not create the Kubernetes resources for using PCC. • **true**: Creates the Kubernetes resources for using PCC. This is the default.
imageName	**ocpcc.pcc.deployment.app**	The name of the PCC image, such as **oracle/pcc**.
imageTag	**ocpcc.pcc.deployment.app**	The tag associated with the image. This is generally the patch set number, prefixed with a colon (:). For example, :**16.26.0.0.0.0**
keystoreAlias	**ocpcc.pcc.configEnv**	The private key alias of the KeyStore.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-9 (Cont.) Pipeline Configuration Center Keys

Key	Path in Values.yaml File	Description
dbSSLMode	**ocpcc.pcc.configEnv**	The type of connection required to connect to the database: • **TWO_WAY**: Two-way SSL authentication is required. In this case, both the client and server must authenticate each other's identity. • **ONE_WAY**: One-way SSL authentication is required. In this case, the client must authenticate the server's identity. This is the default. • **NO**: SSL authentication is not required.
dbWalletType	**ocpcc.pcc.configEnv**	The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**.
extDBSSLWalletSecret **extKeystoreSecret**	**ocpcc.pcc.configEnv**	The names of the pre-created Kubernetes Secrets for the Pipeline Configuration Center KeyStore certificates and wallets. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
tlsVersions	**ocpcc.pcc.configEnv**	The list of TLS versions to support for connection with the WebLogic domain. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**.
login	**ocpcc.pcc.infranet.user**	The username of the service that has permission to access BRM.
serviceType	**ocpcc.pcc.infranet.user**	The POID type of the service that has permission to access BRM.
serviceID	**ocpcc.pcc.infranet.user**	The POID ID of the service that has permission to access BRM.
minSize	**ocpcc.pcc.infranet.connectionpool**	The minimum size of the connection pool.
maxSize	**ocpcc.pcc.infranet.connectionpool**	The maximum size of the connection pool.
loglevel	**ocpcc.pcc.infranet**	The log level for the infranet properties.
addOnProperties	**ocpcc.pcc.infranet**	Empty by default, you can use this key to specify custom infranet properties.
domainUID	**ocpcc.pcc.wop**	The name of the domain. The default is pcc-domain.
adminChannelPort	**ocpcc.pcc.wop**	The NodePort where the admin-server's HTTP service will be accessible. The default is blank. **Note**: Set this key only if you want the **pcc-domain-admin-server-ext** service to deploy as NodePort.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-9 (Cont.) Pipeline Configuration Center Keys

Key	Path in Values.yaml File	Description
serverStartPolicy	**ocpcc.pcc.wop**	The WebLogic servers that the Operator starts when it discovers the domain: • **NEVER**: Does not start any server in the domain. • **ADMIN_ONLY**: Starts only the administration server (no managed servers will be started). • **IF_NEEDED**: Starts the administration server and clustered servers up to the replica count.
isEnabled	**ocpcc.pcc.monitoring**	Whether to enable monitoring of PCC.
nodeSelector	**ocpcc.pcc**	The node selector rules for scheduling WebLogic Server pods on particular nodes using simple selectors.
affinity	**ocpcc.pcc**	The affinity rules for scheduling WebLogic Server pods on particular nodes using more powerful selectors.

Table 7-10 lists the secret keys that directly impact PCC deployment. These keys hold sensitive data and must be handled carefully with controlled access to the file containing its values. See "Secrets" in *Kubernetes Concepts*.

Add these secret keys to your **override-values.yaml** file, and encode all of their values in Base64.

> **Note**
>
> - You can encode strings in Linux by using this command:
>
>   ```
>   echo -n 'string' | base64
>   ```
>
> - You can decode strings in Linux by using this command:
>
>   ```
>   echo 'encoded_string' | base64 --decode
>   ```

Table 7-10 Pipeline Configuration Center Secret Keys

Key	Description
ocpcc.pcc.secretVal.adminPassword	The WebLogic Server administrative password encoded in Base64.
ocpcc.pcc.secretVal.walletPassword	The PCC wallet password encoded in Base64.
ocpcc.pcc.secretVal.rcuSysDBAPassword	The Database Administrator password encoded in Base64.
ocpcc.pcc.secretVal.rcuSchemaPassword	The password for schemas of Oracle Fusion Middleware products that will be created by RCU, which is used by OPSS. The value must be Base64-encoded.
ocpcc.pcc.secretVal.keystoreIdentityPassword	The KeyPass of Identity Keystore, which is used for setting up the SSL-enabled domain. The value must be Base64-encoded.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 7-10 (Cont.) Pipeline Configuration Center Secret Keys

Key	Description
ocpcc.pcc.secretVal.keystoreKeyPassword	The StorePass of Identity Keystore, which is used for setting up the SSL-enabled domain. This value must be Base64-encoded.
ocpcc.pcc.secretVal.keystoreTrustPassword	The StorePass of the Trust Keystore, which is used for setting up the SSL-enabled domain. This value must be Base64-encoded.
ocpcc.pcc.secretVal.pccUserPassword	The PCC user password encoded in Base64.

About PCC Volume Mounts

The PCC container requires Kubernetes volume mounts for sharing the domain and application file system between the WebLogic Cluster servers. There is one volume for the domain. By default, these are created dynamically, using the provisioner defined in BRM, in the **storage-class** key in **oc-cn-op-job-helm-chart**.

To change the volume type or provider, modify the following keys in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

- **ocpcc.pcc.volume.domain.createOption** for the domain file system for PCC.

Creating a WebLogic Domain and Installing the PCC Application

The WebLogic domain is created by a Kubernetes Deployment when **oc-cn-op-job-helm-chart** is installed. The same job also installs the PCC application and deploys the application EAR file onto the WebLogic Cluster.

The **oc-cn-op-job-helm-chart** chart also:

- Creates a Kubernetes ConfigMap and Secrets, which are used throughout the life-cycle of the WebLogic domain.
- Initializes the **PersistentVolumeClaim** for the domain and application file system as well as third-party libraries.

> **Note**
>
> The **override-values.yaml** file that you use for this chart must include BRM override values.

After you install **oc-cn-op-job-helm-chart**, wait until the Kubernetes deployment has reached the **1/1 Running** status. Then, you can install or upgrade **oc-cn-helm-chart** for PCC services.

After the deployment is running, don't delete the chart. Its resources will be used for starting and stopping the servers through **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Setting Up SSO for Pipeline Configuration Center

SSO allows users to log in to applications using a single user name and password combination. You set up SSO for Pipeline Configuration Center cloud native services by using SAML 2.0.

To set up SSO for Pipeline Configuration Center:

1. Export the SAML 2.0 metadata XML file from your identity and access management (IAM) system.

   For example, if you are using Oracle Access Management, you can export the file by following the instructions in "Exporting Metadata" in *Oracle Fusion Middleware Administering Oracle Access Management*.

2. Add the metadata XML file to your BRM cloud native deployment by doing one of the following:

   - Rename the metadata XML file to **metadata.xml**, and then move **metadata.xml** to the **oc-cn-op-job-helm-chart/pcc/idp** directory.
   - Pre-create the IDP metadata ConfigMap for Pipeline Configuration Center and set the **ocpcc.pcc.configEnv.extMetadataCM** key in your **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

     For more information, see "Managing Wallet and KeyStore Certificates" in *BRM Cloud Native System Administrator's Guide*.

3. Configure the KeyStores needed by SAML by doing one of the following:

   - Generate the Identity and Trust KeyStores and then move your files, such as **identity.p12** and **trust.p12**, under the **oc-cn-op-job-helm-chart/pcc/keystore** directory.
   - Pre-create the Kubernetes Secret for the Identity and Trust KeyStores and set the **ocpcc.pcc.configEnv.extKeystoreSecret** key in your **override-values.yaml** file for both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**.

     For more information, see "Managing Wallet and KeyStore Certificates" in *BRM Cloud Native System Administrator's Guide*.

4. In your **override-values.yaml** file for **oc-cn-helm-chart**, set the **isSSOEnabled** key to **true**.

5. In your **override-values.yaml** file for **oc-cn-op-job-helm-chart**, set the following keys:

   - **ocpcc.pcc.configEnv.isSSOEnabled**: Set this to **true**.
   - **ocpcc.pcc.configEnv.keystoreAlias**: Set this to the private key alias of the KeyStore.
   - **ocpcc.pcc.configEnv.keystoreType**: Set this to the file type of the SSL Identity and Trust store, which is either **PKCS12** or **JKS**. The default is **PKCS12**.
   - **ocpcc.pcc.configEnv.keystoreIdentityFileName**: Set this to the name of the Identity KeyStore file.
   - **ocpcc.pcc.configEnv.keystoreTrustFileName**: Set this to the name of the Trust KeyStore file.
   - **ocpcc.pcc.configEnv.samlAsserterName**: Set this to the name of the SAML Asserter. The default is **samlPCCAsserter**.
   - **ocpcc.pcc.configEnv.ssoPublishedSiteURL**: Set this to the base URL that is used to construct endpoint URLs. This is typically the load balancer host and port at which

Copyright © 2019, 2026, Oracle and/or its affiliates.

the server is visible externally. It must be appended with **/saml2**. For example: **https://***LoadBalancerHost***:***LoadBalancerPort***/saml2**.

- **ocpcc.pcc.configEnv.ssoDefaultURL**: Set this to the URL where unsolicited authentication responses are sent if they do not contain an accompanying target URL.
- **ocpcc.pcc.secretVal.keystoreIdentityPassword**: Set this to the StorePass for the Identity KeyStore.
- **ocpcc.pcc.secretVal.keystoreKeyPassword**: Set this to the KeyPass for the Identity KeyStore.
- **ocpcc.pcc.secretVal.keystoreTrustPassword**: Set this to the StorePass for the Trust KeyStore.

6. Configure your load balancer's rules to send responses to the Pipeline Configuration Center WebLogic domain with **/saml2** appended to the URL path.

> **Note**
>
> Add this rule to your existing load balancer rules for routing responses to Pipeline Configuration Center (**/pcc**), the load balancer host name, and so on.

See "Exposing BRM Services Outside the Kubernetes Cluster".

7. Deploy your Pipeline Configuration Center cloud native services by following the instructions in "Deploying BRM Cloud Native Services".
8. After Pipeline Configuration Center is deployed, retrieve the **sp-metadata-admin-server.xml** file from the **/shared/domains/***domainUID* directory in your container, where *domainUID* is the name of your Pipeline Configuration Center domain specified in the **ocpcc.pcc.wop.domainUID** key.

   The XML file configures the Web SSO Provider Partner. It contains the partner's KeyStore certificates, SAML assertion details, and the URLs where the SAML Identity Provider redirects to provide access to Pipeline Configuration Center.
9. Create a profile for your identity provider partner by loading the **sp-metadata-admin-server.xml** file into your IAM system.

   For example, if you are using Oracle Access Management, you can load the file by following the instructions in "Creating Remote Identity Provider Partners" in *Oracle Fusion Middleware Administering Oracle Access Management*.

Setting Up Local Users and Groups for PCC

You have the option to customize the values for **oc-cn-op-job-helm-chart** to create users and groups locally in Oracle WebLogic Server. This would be especially useful for test environments where you might not have Identity Providers or LDAPs available. The groups for the admin user for WebLogic Server cannot be modified using this procedure.

Any passwords must be encoded using Base64. You can leave the password blank, but then the user will not be able to log in to the application directly.

To set up local users and groups for PCC, define the keys under **ocpcc.pcc.wlsUserGroups** in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

A group has to be the Config Admin to access the PCC UI. Only a user associated with the Config Admin group has full access to the PCC user interface. For example:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
Add users and groups to domain's DefaultAuthenticator (local)
        wlsUserGroups:
            groups:
            - name: Config Admin
              description: PCC Admin
            # Each element for this takes "name", "description", "password" (base64
encoded) and list of "groups" that he is part of, like:
            # - name:
            # description:
            # password:
            # groups:
            # - "Regular CSR"
            users:
            - name: pccuser
              description: pccuser
              password: QzFnMmIzdTQj
              groups:
              - "Config Admin"
```

Starting and Stopping WebLogic Servers

When you install **oc-cn-op-job-helm-chart**, the default configuration sets up a WebLogic Cluster with five Managed Servers. When you install or upgrade **oc-cn-helm-chart** for the PCC service, two of the Managed Servers and one Admin Server are started.

By modifying the **override-values.yaml** file for **oc-cn-helm-chart**, you can control:

- The total number of Managed Servers and the initial server start up by using the **totalManagedServers** and **initialServerCount** keys.
- Whether the servers are started or stopped by using the **serverStartPolicy** key. To start the Admin Servers and the Managed Servers in a Cluster, set the key to **IF_NEEDED**. To stop all servers, set the key to **NEVER**.

> **Note**
>
> The keys in the **override-values.yaml** file should be the same as the ones used in **oc-cn-op-job-helm-chart** for keys that are common in both charts.

After you modify the **override-values.yaml** file, update the Helm release for the changes to take effect:

helm upgrade *BrmReleaseName* **oc-cn-helm-chart --values** *OverrideValuesFile* **--namespace** *BrmNameSpace*

where:

- *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.
- *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.
- *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Configuring SSL in PCC

To access PCC over the HTTPS port, SSL must be enabled in the WebLogic domain where PCC is deployed. The BRM cloud native deployment package takes care of the configuration necessary to equip the WebLogic domain with SSL access.

To complete the configuration for SSL setup:

1. Copy PKCS12 files with valid certificates to the **oc-cn-op-job-helm-chart/pcc/keystore** directory:
   - **identity.p12:** Provides the certificate to identify the server.
   - **trust.p12:** Establishes trust for the certificate.

   If your KeyStore files have different file names or file types, such as JKS, override the **keyStoreIdentityFileName**, **keyStoreTrustFileName**, and **keyStoreType** keys in the **override-values.yaml** file for **oc-cn-helm-chart**.

   The **keystoreAlias** key is also mandatory along with **keyStoreIdentityFileName**, **keystoreTrustFileName** to enable SSL.

Copyright © 2019, 2026, Oracle and/or its affiliates.

8

Configuring REST Services

Learn how to integrate external applications with your Oracle Communications Billing and Revenue Management (BRM) cloud native environment by using the BRM and PDC REST services.

Topics in this document:

- Configuring BRM REST Services Manager
- Configuring PDC REST Services Manager

Configuring BRM REST Services Manager

You use BRM REST Services Manager to integrate an external customer experience application with BRM. This allows you to manage billing and rating in BRM and then view your customers' account balances and bills in your external client. For more information, see *REST API Reference for Billing and Revenue Management*.

To configure BRM REST Services Manager in BRM cloud native:

1. Generate an SSL certificate. See "Generating an SSL Certificate for BRM REST Services Manager".
2. Optionally, configure the BRM REST Services Manager SDK. See "Configuring the SDK (Optional)".
3. If BRM and REST Services Manager are located in separate clusters, connect BRM REST Services Manager to BRM. See "Connecting to a Separate BRM Cluster".
4. Override the BRM REST Services Manager-specific keys in the **values.yaml** file. See "Adding BRM REST Services Manager Keys".
5. Optionally, configure policies for authorizing calls to the BRM REST Services Manager REST API endpoints. See "Configuring Policies for API Authorization".
6. Optionally, override the default mappings between BRM REST Services Manager JSON payload objects and BRM opcode flists. See "Configuring and Adding Custom Mapper Files".
7. Optionally, the Kafka notification framework for BRM REST Services Manager. See "Publishing Notifications for External Applications".

Generating an SSL Certificate for BRM REST Services Manager

The following shows the steps for generating a sample SSL certificate:

1. Create a directory named **rsm_keystore** under the **oc-cn-helm-chart/rsm** directory.
2. Generate an SSL certificate. For example:

   ```
   openssl req -x509 -newkey rsa:4096 -keyout openSSLKey.pem -out cert.pem -days 365 -nodes
   ```

3. Generate a **PKCS12** KeyStore file. For example, this creates a KeyStore file named **keystore.p12**:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
openssl pkcs12 -export -out keyStore.p12 -inkey openSSLKey.pem -in cert.pem
```

4. Copy your SSL certificate file to the **oc-cn-helm-chart/rsm/rsm_keystore** directory.

Configuring the SDK (Optional)

To integrate the SDK with BRM REST Services Manager, generate an SDK image as follows:

1. Copy your extended SDK JAR **oc-cn-docker-files-16.26.0.*x*.*x*.*x*/oc-cn-docker-files/ocrsm/brm_rest_services_manager/SDK/libs** to the **oc-cn-docker-files-16.26.0.*x*.*x*.*x*/oc-cn-docker-files/ocrsm/brm_rest_services_manager/SDK** directory.

> **Note**
>
> The SDK JAR can be used directly from **oc-cn-docker-files-16.26.0.*x*.*x*.*x*/oc-cn-docker-files/ocrsm/brm_rest_services_manager/SDK/libs** if no changes are required. If you need to make further customizations, follow the instructions in *REST API Reference for Billing and Revenue Management* and then copy the updated SDK JAR to the **oc-cn-docker-files-16.26.0.*x*.*x*.*x*/oc-cn-docker-files/ocrsm/brm_rest_services_manager/SDK** directory.

2. In your **override-values.yaml** file for **oc-cn-helm-chart**, set the **ocrsm.rsm.configEnv.rsmExtensionJar** key to the name of your extended SDK JAR file, such as **BRMRESTExtension.jar**.
3. Go to the **oc-cn-docker-files-16.26.0.*x*.*x*.*x*/oc-cn-docker-files/ocrsm/brm_rest_services_manager/SDK** directory.
4. Build the Podman image by running this command:

```
podman build --format docker --tag imagerepo/brm-rest-services-manager-extension:1 .
```

5. Push the SDK image to the repository by running this command:

```
podman login --username user --password password imagerepo
podman push imagerepo/brm-rest-services-manager-extension:1
```

Connecting to a Separate BRM Cluster

If BRM is located in a separate cluster from BRM REST Services Manager, do the following to connect BRM REST Services Manager to BRM:

1. Open the **configmap_env_brmrsm.yaml** file in a text editor.
2. Set **BRM_HOST_NAME** to the host name of the cluster on which BRM is located. The default value is **cm**.
3. Save and close the file.

Adding BRM REST Services Manager Keys

Table 8-1 lists the keys that directly impact BRM REST Services Manager. Add these keys to your **override-values.yaml** file with the same path hierarchy.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Keys with the path **ocrsm.rsm.secretVal** hold sensitive data. Handle them carefully with controlled access to the override file containing their values. Encode all of these values in Base64. See "Secrets" in *Kubernetes Concepts*.

Table 8-1 BRM REST Services Manager Keys

Key	Path in Values.yaml File	Description
isEnabled	**ocrsm.rsm**	Whether to deploy BRM REST Services Manager with BRM cloud native (true) or not (false). The default is **true**.
labels.*	**ocrsm.rsm**	The string used to form the names of BRM REST Services Manager.
deployment.*	**ocrsm.rsm**	The details for deploying BRM REST Services Manager. • **deadlineSeconds**: The maximum time, in seconds, for a deployment to make progress before it is considered failed. The default is **120**. • **revisionHistLimit**: The maximum number of old ReplicaSets for this deployment to retain. The remaining is garbage-collected in the background. The default is **10**. • **imageName**: The name of the BRM REST Services Manager image, such as **oracle/brm-rest-services-manager**. • **imageTag**: The tag associated with the image, such as **:16.26.0.0.0.0**.
sdk.*	**ocrsm.rsm.deployment**	The details about the BRM REST Services Manager SDK. • **imageName**: The name of the BRM REST Services Manager SDK image. • **imageTag**: The tag associated with the BRM Services Manager SDK image, such as **:1**.
probe.ready.*	**ocrsm.rsm.deployment**	The configuration for the readiness probe. • **delayInSec**: The duration, in seconds, to wait before performing the first readiness probe. The default is **30**. • **intervalInSec**: How often, in seconds, to perform the readiness probe. The default is **5**. • **maxAttempts**: The maximum number of consecutive failures before the probe is considered failed. The default is **15**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-1 (Cont.) BRM REST Services Manager Keys

Key	Path in Values.yaml File	Description
REST Services Manager Volumes	**ocrsm.rsm.deployment**	The details about the REST Services Manager volumes. • **volMntKeyStore.name**: The volume containing the BRM REST Services Manager SSL keystore certificate. • **volMntLogs.name**: The name of the log volume mount where the SDK JAR is mounted. • **volMntSDK.name**: The name of the SDK volume mount where the SDK Jar is mounted. • **volMntSecretEnv.name**: The name of the volume mount that holds all passwords as a secret. • **volMntSecurity.***: The name and path of the volume mount containing the server security. • **volMntAppExternalProperties.***: The name and path of the volume mount containing the application's external properties.
configEnv.*	**ocrsm.rsm**	The configuration details for the BRM REST Services Manager API. • **name**: The name of the BRM REST Services Manager API ConfigMap. The default is **brm-rest-services-manager-env-configmap**. • **httpPort**: The HTTP port in the container on which to deploy BRM REST Services Manager. The default is **9090**. • **httpsPort**: The HTTPS port in the container on which to deploy BRM REST Services Manager. The default is **8080**. • **adminPort**: The administration port for health, metrics, and other administration-related activities. The default is **9060**. • **tlsVersions**: The list of TLS versions to support for connection with the WebLogic domain. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**. • **rsmCertificateFileName**: The SSL certificate file name for BRM REST Services Manager. • **trustStoreFileName**: This is the optional file name for the truststore. Set this key if the default Java truststore needs to be overridden.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-1 (Cont.) BRM REST Services Manager Keys

Key	Path in Values.yaml File	Description
infranet.*	**ocrsm.rsm.configEnv**	The details for connecting to BRM cloud native. • **user.login**: The login name of the service with permissions to access BRM. • **user.serviceType**: The POID type for the service having permissions to access BRM. • **user.serviceId**: The POID of the service having permissions to access BRM. • **connectionpool.minSize**: The minimum number of threads in the connection pool. • **connectionpool.maxSize**: The maximum number of threads in the connection pool. • **PcmTimeoutInMsecs**: The duration, in milliseconds, the PCM waits before timing out. • **login.type**: Whether a login name and password are required to connect to the BRM database (**1**) or not (**0**). The default is **1**.
brmSSLWalletFileName	**ocrsm.rsm.configEnv**	The BRM SSL wallet file name. The default is **cwallet.sso**.
rsmExtensionJar	**ocrsm.rsm.configEnv**	The file name of the BRM REST Service Manager SDK JAR, such as **BRMRESTExtension.jar**.
bipURL	**ocrsm.rsm.configEnv**	The Oracle Analytics Publisher URL.
bipUserId	**ocrsm.rsm.configEnv**	The Oracle Analytics Publisher user ID.
Zipkin Tracing Details	**ocrsm.rsm.configEnv**	The Zipkin tracing details. • **isTracingEnabled**: Whether to enable tracing for the BRM REST Services Manager API (**true**) or not (**false**). The default is **false**. • **zipkinHostName**: The host name of the Zipkin tracing collector. • **zipkinPort**: The port of the Zipkin tracing collector. • **zipkinProtocol**: The protocol of the Zipkin tracing collector, such as **http** or **https**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-1 (Cont.) BRM REST Services Manager Keys

Key	Path in Values.yaml File	Description
BRM REST Services Manager Cache Details	**ocrsm.rsm.configEnv**	The cache details. • **baseURL**: The base URL with resource details to return in the response of BRM REST Services Manager requests. **Note:** After deployment, you can update this value by editing your **override-values.yaml** file and then doing a Helm upgrade. • **cacheEnabled**: Whether BRM REST Services Manager supports an internal cache (**true**) or not (**false**). The default is **false**. • **cacheType**: The cache type: **private** or **public**. The default is **private**. • **cacheMaxAge**: The time, in seconds, that the response remains fresh in cache after the response is generated. The default is **86400**. • **cacheServices**: The services that need to provide cache support. The default is **describe**, which is the resource locator for the Describe API in the REST API path.
securityEnabled	**ocrsm.rsm.configEnv**	The flag to indicate if token-based authentication is enabled for BRM REST Services Manager (**true**) or not (**false**). The default is **true**.
oidc.*	**ocrsm.rsm.configEnv**	The Identity Provider (IdP) authentication details. • **identity-uri**: The URI of the Identity Server, used as the base URL to retrieve metadata from the Identity Server. • **client-id**: The client ID generated by the Identity Server, used to validate the token. • **client-secret**: The client secret generated by the Identity Server, used to authenticate the application when requesting a JWT based on a code. Do not set this key directly. Instead, specify the client secret password under the **secretVal.clientSecret** key. • **proxyHost**: The proxy host of the IdP, if defined. When set, this triggers the use of a proxy for HTTP requests. • **scope-audience**: The audience for the scope required by this application. This is prefixed to the scope name when requesting scopes from the Identity Server. • **audience**: The secondary audience configured in the IdP. If no secondary audience is configured, use the primary audience, which is the same as the scope-audience. • **introspect-endpoint-uri**: The endpoint URI used to validate the JWT.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-1 (Cont.) BRM REST Services Manager Keys

Key	Path in Values.yaml File	Description
roleMapperName	**ocrsm.rsm.configEnv**	To authorize requests, specify a mapper based on your Identity Provider (IdP): use **idcs-role-mapper** for Oracle IDCS or **oam-role-mapper** for Oracle Access Manager. This step is necessary when your JSON Web Tokens (JWTs) do not adhere to the MicroProfile JWT RBAC v2.1 specification and lack a "groups" claim because it enables fetching user/client groups and roles. If your JWTs conform to the JWT RBAC v2.1 specification, leave this key empty.
oam.*	**ocrsm.rsm.configEnv**	The Oracle Access Manager authentication details. These keys ensure backward compatibility for users who have not yet migrated to the latest versions of Oracle Access Management, where roles and groups are included in the JWT token. **Note**: This section is enabled only if **roleMapperName** is set to **oam-role-mapper**. • **oudHostName**: The Oracle Unified Directory host name. • **oudRootUserDN**: The Oracle Unified Directory root user domain name. • **oudHttpPort**: The Oracle Unified Directory HTTP port. • **oudHttpsPort**: The Oracle Unified Directory HTTPS port. • **oudUserBaseDN**: The Oracle Unified Directory user domain name. • **oudGroupDN**: The Oracle Unified Directory group domain name. • **msgType**: The message type based on the schema used to search roles in the Oracle Unified Directory. • **filter**: The filter based on the user attribute.
Log Level Details	**ocrsm.rsm.configEnv**	The logging level. The possible values for these keys are: **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, **FINEST**. • **logLevel**: The application log level. • **helidonSecurityLogLevel**: The Helidon security log level. • **helidonWebServerLogLevel**: The Helidon WebLogic server log level. • **helidonConfigLogLevel**: The Helidon configuration log level. • **helidonMicroProfileLogLevel**: The Helidon MP log level. • **helidonCommonLogLevel**: The Helidon common log level. • **nettyServerLogLevel**: The embedded netty server log level. • **jerseyLogLevel**: The Jersey log level. • **jbossWeldLogLevel**: The Helidon JBossWeld log level. • **auditLogLevel**: The audit log level.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-1 (Cont.) BRM REST Services Manager Keys

Key	Path in Values.yaml File	Description
jvmOpts	**ocrsm.rsm.configEnv**	The Java options to configure when setting resources for the containers, such as heap memory. For example: -XX:InitialRAMPercentage=25.0 -XX:MaxRAMPercentage=50.0.
extKeystoreSecret	**ocrsm.rsm.secretKeyStore**	The name of the external Kubernetes secret containing the Identity and Trust files. **Note:** Override the **rsmCertificateFileName**, **trustStoreFileName**, and **keyStoreFileName** with the respective names of the truststore and keystore. See "Secrets" in *Kubernetes Concepts*, and "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
secretVal.*	**ocrsm.rsm**	The credentials for accessing the system. • **name**: The name of the Kubernetes secret that copies certificates to the container. The default is **brm-rest-services-manger-env-secret**. • **rsmCertificatePassword**: The Base64-encoded certificate password for BRM REST Services Manager. • **brmInfranetWalletPassword**: The Base64-encoded wallet password. You can use any password. This password will be used to store the Oracle Analytics Publisher and Infranet connections in the wallet and can be used to access the same. • **bipPassword**: The Base64-encoded Oracle Analytics Publisher password. • **clientSecret**: The Base64-encoded IDCS client secret. • **trustStorePassword**: The truststore file password.
secret.*	**ocrsm.rsm**	The details about BRM REST Services Manager secrets: • **extBrmRsmPasswordSecret**: The name of the external secret that stores all passwords. • **extBrmRestServicesManagerSslSecret**: The name of the external secret that copies certificates to the container. • **extBrmRsmProfileEnvSecret**: The name of the external secret that stores all profile passwords. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
affinity	**ocrsm.rsm**	The rules for scheduling pods on particular nodes using more powerful selectors using affinity rules.
nodeSelector	**ocrsm.rsm**	The rules for deploying the pod on specific nodes.
addOnPodSpec	**ocrsm.rsm**	The details for extending pod specification or overriding features. By default, this key is empty.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-1 (Cont.) BRM REST Services Manager Keys

Key	Path in Values.yaml File	Description
resources.*	**ocrsm.rsm**	The minimum and maximum CPU and memory resources that containers can use. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
hpa.*	**ocrsm.rsm**	The details for scaling up or down the number of pod replicas in your deployment based on a pod's CPU or memory utilization. By default, the Horizontal Pod Autoscaler is disabled. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*.
service.*	**ocrsm.rsm**	The brm-rest-services-manager service's details. • **name**: The name of the service: **brm-rest-services-manager**. • **type**: The service type. The default is **ClusterIP**.
mapperConfiguration	**ocrsm.rsm**	This contains the Mapper Configuration keys which are required for the APIs and events that use mapper framework. See "Configuring and Adding Custom Mapper Files" for more information.
consumers.*	**ocrsm.rsm**	The consumer configuration details for consuming events from Kafka topics and database queues. **Note:** Uncomment or add the required configurations to work with event notifications.
producer.*	**ocrsm.rsm**	The Kafka connection settings required for the producer to publish event messages. **Note:** Uncomment or add the required configurations in order to work with event notifications.
connection-profiles.*	**ocrsm.rsm**	The connection profile settings required for secure connections to Kafka and the database. **Note:** Uncomment or add the required configurations in order to work with event notifications

You can use the following commands to encode and decode passwords in Base64 format:

- To encode strings in Linux:

  ```
  echo -n 'password' | base64
  ```

- To decode strings in Linux:

  ```
  echo 'encoded_password' | base64 --decode
  ```

Configuring Policies for API Authorization

To configure the policies for API authorization:

1. Define the API authorization rules in a policy file.

   You can use the sample authorization policy ConfigMap (**configmap_auth_policy_brmrsm.yaml**) as a template for defining API authorization rules.

Copyright © 2019, 2026, Oracle and/or its affiliates.

2. For any new BRM REST Services Manager API endpoints, ensure that appropriate policy statements are added to the file. This is essential for enforcing proper authorization and access restrictions for each new API.
3. Run the **helm upgrade** command to update the Helm release:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
   ```

Configuring and Adding Custom Mapper Files

By default, BRM REST Services Manager uses predefined mapper files to translate between JSON payload objects and BRM opcode flists for the latest versions of account and inventory management. However, you can customize this mapping by creating and adding your own mapper files. This allows you to extend or override the default mappings to meet your specific integration needs.

To introduce custom mapping logic for REST API payloads, extend or override the mapping between BRM REST Services Manager JSON objects and BRM opcode flists:

1. Draft your customization as one or more YAML files. Follow the same structure as the sample files in **oc-cn-helm-chart/rsm/referenceMappers**.

   The following example shows how to map a custom payload field (PIN_FLD_CUSTOM) for the Create a Product REST API request:

   ```
   mapper:
       custom:
           v5:
               tmf637:
                   productCreateEvent:
                      product:
                        request:
                          jsonCustomField:
                                field: PIN_FLD_CUSTOM_FIELD
                                fieldValue: ""
                                optional: false
                                validation: "enum:[inactive, active, cancelled]"
   ```

2. Add your custom YAML files to the BRM REST Services Manager deployment by doing one of the following:
   - Create a ConfigMap from a single YAML file:

     ```
     kubectl create configmap ConfigMapName --from-file=YamlFileName -n BrmNameSpace
     ```

     where:
     - *ConfigMapName* is the name of the ConfigMap to create.
     - *YamlFileName* is the path to your custom YAML file.
     - *BrmNameSpace* is the BRM Kubernetes namespace.
   - Create a ConfigMap from all YAML files in a directory:

     ```
     kubectl create configmap ConfigMapName --from-file=. -n BrmNameSpace
     ```

     > **Note**
     >
     > Run all commands from the directory containing the file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Add your custom YAML files to the BRM REST Services Manager deployment without pre-creating a ConfigMap. Place files of size upto 1 MiB in the **oc-cn-helm-chart/rsm/extendedMappers**. Place larger files in **SDK/extendedMappers** within the SDK project, and supply the SDK image to REST Services Manager. Use the existing REST Services Manager packaging steps to build and provide the SDK image to RSM.

> **Note**
>
> All methods are for mapper files less than 1MiB. You can use SDK to create custom mapper files larger than 1 MiB.

3. Provide the path from your custom mapper files as requests, responses, or execute respectively. Update the **override_values.yaml**:

```
ocrsm:
     rsm:
        mapperConfiguration:
            config:
              paths:
                event:
                  v5:
                    ProductCreateEvent:
                      mapper-specification-keys:
                        default: Product
                        Event:
                          execute:
                            - "mapper.brm.v5.tmf637.productCreateEvent.execute"
                          request:
                            - "mapper.brm.v5.tmf637.productCreateEvent.request"
                          response:
                            - "mapper.brm.v5.tmf637.productCreateEvent.response"
                            - "mapper.custom.v5.tmf637.productCreateEvent.response"
                        Product:
                          request:
                            - "mapper.brm.v5.tmf637.post.request.product"
                            - "mapper.brm.v5.tmf637.post.request.extendedProduct"
                            - "mapper.custom.v5.tmf637.post.request.product"
                          response:
                            - "mapper.brm.v5.tmf637.post.response.product"
                            - "mapper.brm.v5.tmf637.post.response.extendedProduct"
                            - "mapper.custom.v5.tmf637.post.response.product"
```

where:

- **ProductCreateEvent** defines the type of event.
- **mapper-specification-keys** specifies logical profiles or mapping variants.
- **default** is used when no explicit profile is provided.
- **execute**, **request**, and **response** designate the mapping configuration files to be applied for execution opcode, request transformation, and response transformation respectively.

4. Update the following keys in your **override_values.yaml** file for **oc-cn-helm-chart**:
   - **ocrsm.rsm.externalExtendedMapper**: If you created a ConfigMap, set this to the name of the ConfigMap you created in step 2.
   - **ocrsm.rsm.configEnv.restartVersion**: Increment this value by one and run the helm upgrade to force restart and initialize the changes.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> For more information on Mapper keys, refer to Table 8-1.

5. Run the **helm upgrade** command for **oc-cn-helm-chart**:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
   ```

Publishing Notifications for External Applications

Use the BRM REST Services Manager notification framework to publish TM Forum (TMF) events to Apache Kafka topics so downstream applications, such as Siebel CRM or Order and Service Management (OSM), can retrieve and process them.

The framework publishes notifications when any of the following occur in BRM:

- A billing account is created or deleted.
- A billing account's state or attributes have changed.
- A party account is created or deleted.
- A party account's state or attributes have changed.
- A product is created or deleted.
- A product's state or attributes have changed.

For details about the notification event content, see *REST API Reference for Billing and Revenue Management*.

Before you begin, ensure that your system meets the notification framework prerequisites. See "Notification Framework Prerequisites".

To configure BRM REST Services Manager to publish notifications:

1. Open your **override-values.yaml** file for **oc-cn-helm-chart**.
2. Configure the connection profiles for Kafka brokers and the Oracle AQ database by uncommenting and setting the keys under **ocrsm.rsm.connection-profiles**. See "Setting Up Connection Profiles".
3. Configure one or more consumers to read messages from Kafka topics by uncommenting and setting the keys in **ocrsm.rsm.consumers**. See "Configuring Consumers for Kafka Topics".
4. Configure one or more consumers to read messages from the Oracle AQ database queue by uncommenting and setting the keys under **ocrsm.rsm.consumers**. See "Configuring Consumers for Database Queues".
5. Configure one or more producers to write messages to Kafka topics by uncommenting and setting the keys under **ocrsm.rsm.producer**. See "Setting Up Producers".
6. Store secrets in Base64-encoded format in the **override-values.yaml** file so they can be added to the wallet at runtime.
7. Save the configuration and verify that all folder paths referenced in the **override-values.yaml** file exist and are accessible to the REST Services Manager process.
8. Validate that topics, partitions, security settings, and service connections are correct. See "Performing Verification and Sanity Checks".

Copyright © 2019, 2026, Oracle and/or its affiliates.

9. Use the **helm upgrade** command to deploy and start the REST Services Manager notification services.
10. Validate the service by reviewing logs and performing functional checks. If needed, customize event handling with mapper overrides. Follow the approved restart procedure for the REST Services Manager Notification Service after configuration updates.
11. Confirm that the service connects to the Kafka brokers and joins the expected consumer groups.
12. Restart REST Services Manager cloud native after applying configuration or credential changes.

> **Note**
>
> To apply broker SSL/SASL parameters and store passphrases securely:
>
> - Enter SSL configurations (KeyStore and TrustStore locations and types, and passwords) in **override-values.yaml** using wallet references.
> - Do not store plain-text credentials in configuration files.

Notification Framework Prerequisites

Ensure that your environment satisfies all prerequisites for the REST Services Manager Notification Service:

1. Create your Kafka cluster and topics. Ensure that the configuration aligns with your environment and security standards. To do so, follow the instructions in "Configuring Your AQ Database Queues" in *BRM System Administrator's Guide*.
   - Provision a Kafka cluster with sufficient brokers to meet your throughput and high availability requirements.
   - Define all topics required for the REST Services Manager Notification Service, including source, sink, retry, and dead-letter queue (DLQ) topics.
   - Set the required number of partitions for each topic and apply appropriate access control lists (ACLs).
   - Record the host names and port numbers of all Kafka brokers.
   - Monitor Kafka server logs to validate connectivity and detect errors.
2. Configure the Enterprise Application Integration (EAI) Framework to convert TMF-related data into business events that can be sent to the Oracle Data Manager (DM).
   - Map BRM events to opcodes by using the event notification file. The default file (*BRM_home***/sys/data/config/pin_notify**) includes TMF-related events. Update the event list as needed and load the file into the database. See "Using Event Notification" in *BRM Developer's Guide*.
   - Configure the Payload Generator External Module (EM) to assemble complete business events, generate required data, and send business events to the Oracle DM. The default file (*BRM_home***/sys/eai_js/payloadconfig_crm_sync**) defines TMF-related business events. Update the list as needed. See "Configuring Business Events" in *BRM Developer's Guide*.
3. Create an Oracle Database Wallet for sensitive values.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Configure an Oracle Database Wallet to securely store sensitive credentials, such as SSL/TLS passphrases, KeyStore and TrustStore files, and database schema credentials.
- The wallet contents and certificate requirements depend on the database listener and instance being preconfigured for SSL/TLS. Ensure database encryption is enabled before mapping KeyStore and TrustStore locations in the wallet.
- Store wallet, KeyStore, and TrustStore files in a secure location with restrictive file permissions.
- Do not place passwords or secrets in plain-text configuration files.

Setting Up Connection Profiles

Configure Kafka and database connection profiles by mapping the required parameters and integrating wallet-based secrets. Table 8-2 lists the keys that appear under **ocrsm.rsm.connection-profiles**. Apply the appropriate SSL/SASL parameters to secure access to Kafka brokers.

> **Note**
>
> Store the KeyStore, TrustStore, and wallet files in the **oc-cn-helm-chart/rsm/rsm_keystore** directory. Store secrets in Base64-encoded format in the **override-values.yaml** file so they can be added to the wallet at runtime.

Table 8-2 Connection Profile Keys

Key Name	Description
id	A unique identifier for the connection profile.
profile-type	Whether the profile contains connection information for the Kafka broker (kafka) or database (database).
host	The host name or IP address of the Kafka broker or database.
port	The network port to connect to the Kafka broker or database.
username	The authentication user name, if required for SASL or similar mechanisms (often empty for SSL-only setups).
password	The key that references the Base64-encoded password (for example, stored in the Wallet) for authenticating this profile.
truststore-type	The type of TrustStore file used for SSL/TLS connections: PKCS12 or JKS.
truststore-name	The TrustStore file name that contains the trusted certificate authorities.
truststore-password	The Base64-encoded password key used to retrieve the TrustStore (for example, stored in the wallet) for secure access.
keystore-type	The type of the KeyStore file that holds client SSL certificates: PKCS12 or JKS.
keystore-name	The KeyStore file name that contains the client SSL certificate and private key.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-2 (Cont.) Connection Profile Keys

Key Name	Description
keystore-password	The key used to retrieve the Base64-encoded password (for example, stored in the wallet) for secure access.
properties.*	(Kafka profile only) The Kafka client properties for offsets, timeouts, batching, and so on. Keys and values must follow standard Kafka property names and guidelines. • **auto.offset.reset**: Where to start consuming messages if no offset is committed, such as **latest** to start from the latest message. • **enable.auto.commit**: Whether the Kafka consumer auto-commits offsets (**true**) or requires manual management (**false**). • **session.timeout.ms**: The timeout, in milliseconds, to detect consumer failures. • **heartbeat.interval.ms**: The interval, in milliseconds, for sending heartbeats to the Kafka broker. • **connections.max.idle.ms**: The maximum idle time for a connection before closing. • **max.poll.records**: The maximum number of records returned in a single poll. • **max.poll.interval.ms**: The maximum interval between poll calls before the Kafka broker considers the consumer failed. • **max.partition.fetch.bytes**: The maximum bytes per partition fetched in a request. • **acks**: The number of acknowledgments the producer requires for a request, such as **all** for full commits. • **retries**: The maximum number of retry attempts for failed producer requests. • **batch.size**: The Kafka producer's batch memory size in bytes. • **linger.ms**: The delay, in milliseconds, before sending records to allow batching. • **buffer.memory**: The Kafka producer's total buffer memory in bytes. • **compression.type**: The compression algorithm to use, such as **none** or **gzip**.
properties.service-name	(Database profile only) The specific name of the database service (or SID) for establishing a connection.

The following shows sample settings for a Kafka profile:

```
connection-profiles:
     - id: "cp1"
       profile-type: "kafka"
       host: "host"
       port: port
       username: ""
       password: "base64EncodedPassword"
       truststore-type: PKCS12
       truststore-name: "truststore.p12"
       truststore-password: "base64EncodedPassword"
       keystore-type: PKCS12
       keystore-name: "keystore.p12"
       keystore-password: "base64EncodedPassword"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
  properties:
    auto.offset.reset: latest
    enable.auto.commit: false
    session.timeout.ms: 60000
    heartbeat.interval.ms: 10000
    connections.max.idle.ms: 3600000
    max:
      poll:
        records: 10
        interval.ms: 600000
      partition.fetch.bytes: 1048576
    acks: all
    retries: 3
    batch.size: 16384
    linger.ms: 1
    buffer.memory: 33554432
    compression.type: none
```

The following shows sample settings for a database profile:

```
- id: "cp2"
  profile-type: "database"
  host: "host"
  port: port
  truststore-type: PKCS12
  truststore-name: "truststore.p12"
  truststore-password: "base64EncodedPassword"
  keystore-type: PKCS12
  keystore-name: "keystore.p12"
  keystore-password: "base64EncodedPassword"
  properties:
    service-name: pindb.us.example.com
```

Configuring Consumers for Kafka Topics

Configure Kafka consumers by defining consumer properties, runtime behavior, and topic-event mappings under **ocrsm.rsm.consumers**. Table 8-3 lists the **override-values.yaml** keys for configuring consumers to read messages from Kafka topics.

> **Note**
>
> When using multiple REST Services Manager replicas, ensure the number of partitions for each downstream consumer chain aligns with the parent consumer's partitions to maintain ordering and parallelism.

Table 8-3 Kafka Consumer Keys

Key Name	Description
connection-profile-id	The unique ID for the consumer profile instance.
properties.group.id	The unique ID for the consumer group managing coordination and offsets. **Note**: Use a distinct consumer group ID for each independent consumer. Consumers in the same downstream chain may share a group ID as required.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-3 (Cont.) Kafka Consumer Keys

Key Name	Description
configuration.*	The configuration for the consumer profile instance: • **max-retries**: The maximum number of retry attempts for processing an event before routing it to the retry or DLQ topic. • **retry-delay-interval-ms**: The delay, in milliseconds, between retry attempts after a processing failure. • **record-fetch-timeout-ms**: The maximum time, in milliseconds, the consumer waits for records during a fetch operation. (Also appears as **record.fetch.timeout.ms.**) • **brm-connection-retry-interval-ms**: The interval, in milliseconds, to wait between attempts to reconnect to BRM services. (Also appears as **brm.connection.retry.interval.ms**.) • **strict-ordering**: A flag indicating whether to enforce strict ordering for message keys (**true**) or not (**false**). • **buffer-strategy**: The method used for buffering events within the consumer; this is implementation-specific.
topic-event-config.*	The configuration for a Kafka topic to consume messages from: • **source-topic**: The name of the Kafka topic the consumer reads from. • **sink-topic**: The name of the Kafka topic to which the consumer sends failed messages. • **key-state-topic**: The name of an optional topic used to track and manage the message key states. • **consumer-thread-count**: The number of worker threads allocated for polling messages from the source topic. • **worker-thread-count**: The number of worker threads used for processing polled events. • **events**: A list of event types that the consumer instance will process, such as **ProductCreateEvent** and **ProductStateChangeEvent**.
topic-event-config.downstream-consumers.*	A list of additional consumer stages for retry or DLQ handling, each with its own configuration.

The following shows keys for a sample Kafka consumer named **cp1**:

```
consumers:
     - connection-profile-id: "cp1"
       properties:
         group.id: cgroup-brmrsm-productInventory
       configuration:
         max-retries: 3
         retry-delay-interval-ms: 1000
         record-fetch-timeout-ms: 1000
         brm-connection-retry-interval-ms: 30000
         strict-ordering: false
         buffer-strategy:
       topic-event-config:
         source-topic: productInventory
         sink-topic: productInventoryRetry
         key-state-topic: keyStateTopic
         consumer-thread-count: 1
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
      worker-thread-count: 2
      downstream-consumers:
        - connection-profile-id: "cp1"
          properties:
            group.id: cgroup-brmrsm-productInventory
          configuration:
            max-retries: 3
            retry-delay-interval-ms: 1000
            record.fetch.timeout.ms: 1000
            brm.connection.retry.interval.ms: 30000
          topic-event-config:
            source-topic: productInventoryRetry
            sink-topic: productInventoryDLQ
            consumer-thread-count: 1
            worker-thread-count: 2
            events:
              - ProductCreateEvent
              - ProductAttributeValueChangeEvent
              - ProductStateChangeEvent
              - ProductDeleteEvent
      events:
        - ProductCreateEvent
        - ProductAttributeValueChangeEvent
        - ProductStateChangeEvent
        - ProductDeleteEvent
    - connection-profile-id: "cp1"
      properties:
        group.id: cgroup-brmrsm-partyAccountInventory
      configuration:
        max-retries: 1
        retry-delay-interval-ms: 1000
        record-fetch-timeout-ms: 1000
        brm-connection-retry-interval-ms: 30000
        strict-ordering: false
        buffer-strategy:
      topic-event-config:
        source-topic: partyAccountInventory
        sink-topic: partyAccountInventoryRetry
        key-state-topic: keyStateTopic
        consumer-thread-count: 1
        worker-thread-count: 2
        downstream-consumers:
          - connection-profile-id: "cp1"
            properties:
              group.id: cgroup-brmrsm-partyAccountInventory
            configuration:
              max-retries: 1
              retry-delay-interval-ms: 1000
              record-fetch-timeout-ms: 1000
              brm-connection-retry-interval-ms: 30000
            topic-event-config:
              source-topic: partyAccountInventoryRetry
              sink-topic: partyAccountInventoryDLQ
              key-state-topic:
              consumer-thread-count: 1
              worker-thread-count: 2
              events:
                - PartyAccountCreateEvent
                - PartyAccountAttributeValueChangeEvent
                - PartyAccountStateChangeEvent
                - PartyAccountDeleteEvent
        events:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
      - PartyAccountCreateEvent
      - PartyAccountAttributeValueChangeEvent
      - PartyAccountStateChangeEvent
      - PartyAccountDeleteEvent
  - connection-profile-id: "cp1"
    properties:
      group.id: cgroup-brmrsm-billingAccountInventory
    configuration:
      max-retries: 1
      retry-delay-interval-ms: 1000
      record-fetch-timeout-ms: 1000
      brm-connection-retry-interval-ms: 30000
      strict-ordering: false
      buffer-strategy:
    topic-event-config:
      source-topic: billingAccountInventory
      sink-topic: billingAccountInventoryRetry
      key-state-topic: keyStateTopic
      consumer-thread-count: 1
      worker-thread-count: 2
      downstream-consumers:
        - connection-profile-id: "cp1"
          properties:
            group.id: cgroup-brmrsm-billingAccountInventory
          configuration:
            max-retries: 1
            retry-delay-interval-ms: 1000
            record-fetch-timeout-ms: 1000
            brm-connection-retry-interval-ms: 30000
          topic-event-config:
            source-topic: billingAccountInventoryRetry
            sink-topic: billingAccountInventoryDLQ
            key-state-topic:
            consumer-thread-count: 1
            worker-thread-count: 2
            events:
              - BillingAccountCreateEvent
              - BillingAccountAttributeValueChangeEvent
              - BillingAccountStateChangeEvent
              - BillingAccountDeleteEvent
      events:
        - BillingAccountCreateEvent
        - BillingAccountAttributeValueChangeEvent
        - BillingAccountStateChangeEvent
        - BillingAccountDeleteEvent
```

Configuring Consumers for Database Queues

Table 8-4 lists the **override-values.yaml** keys for configuring consumers to read messages from the Oracle AQ database queue. These keys appear under **ocrsm.rsm.consumers**.

Table 8-4 Consumer Keys for Database Queues

Parameter	Explanation
connection-profile-id	The ID for the database connection profile. This value must exactly match the ID of a profile with **profile-type** set to **database**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-4 (Cont.) Consumer Keys for Database Queues

Parameter	Explanation
properties.*	The properties for connecting to the database queue. • **username**: The login name for authenticating the connection to the database or queue. • **password**: The key used to securely retrieve the Base64-encoded authentication password (such as from a Kubernetes Secret or Oracle Wallet). • **queue-name**: The name of the queue from which this consumer reads messages, such as for queue-based processing.
wallet-name	The name of the wallet or credential store used for secure connections to the database. If provided, wallet credentials take precedence over TrustStores and KeyStore files. Place the wallet in **rsm/rsm_keystore**.
configuration.worker-thread-count	The number of worker threads to process events from the database queue.

The following example shows a Kafka consumer configuration in the *REST_home***/scripts/override-values.yaml**:

```
- connection-profile-id: "cp2"
  properties:
    username: ""
    password: "base64EncodedPassword"
    queue-name: OPEN_API_QUEUE
  wallet-name: db_wallet
  configuration:
    worker-thread-count: 1
```

Setting Up Producers

Configure producers to publish supported TMF business events to one or more Kafka topics. Table 8-5 lists the keys under **ocrsm.rsm.producer**.

Table 8-5 Producer Keys

Parameter	Explanation
connection-profile-id	The name of the connection profile to use.
properties	The Kafka properties to override or producer-specific properties to add.
event-topics-config.*	The business events to publish to each Kafka topic. • **event-name**: The business event type that triggers the producer to publish the message to the Kafka topic, such as **ProductCreateEvent**. • **topic-names**: The list of one or more Kafka topics to which messages for the corresponding event are published.

The following shows keys for a sample Kafka producer named cp1:

```
producer:
     connection-profile-id: "cp1"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
properties:
event-topics-config:
  - event-name: "ProductCreateEvent"
    topic-names: [ "productCreateTopic", "productCreateTopic2" ]
  - event-name: "ProductDeleteEvent"
    topic-names: [ "ProductDeleteTopic" ]
  - event-name: "ProductAttributeValueChangeEvent"
    topic-names: [ "productAttributeValueTopic" ]
  - event-name: "ProductStateChangeEvent"
    topic-names: [ "productStateChangeTopic" ]
  - event-name: "PartyAccountCreateEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
  - event-name: "PartyAccountAttributeValueChangeEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
  - event-name: "PartyAccountStateChangeEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
  - event-name: "PartyAccountDeleteEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
  - event-name: "BillingAccountCreateEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
  - event-name: "BillingAccountAttributeValueChangeEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
  - event-name: "BillingAccountStateChangeEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
  - event-name: "BillingAccountDeleteEvent"
    topic-names: [ "partyAccountNotificationTopic" ]
```

Performing Verification and Sanity Checks

Perform checks to ensure topics, partitions, security settings, and service connections are correct. Verify functional connectivity and end-to-end message flow.

- Confirm that all required topics exist with the correct partition counts and ACLs.
- Ensure that consumers join the correct consumer groups and receive expected partition assignments.
- Verify that SSL or SASL connections to brokers are successfully established and validated.
- Publish test messages and verify end-to-end flow from producers through consumers, including retry and DLQ paths.
- Use provided scripts and logs to verify wallet entry references and KeyStore/TrustStore integrity.

Configuring PDC REST Services Manager

You use PDC REST Services Manager to integrate an enterprise product catalog, such as Oracle Digital Experience for Communications Launch Experience, with PDC. This enables you to create a variety of product offerings in your enterprise product catalog and then have all of the rating and billing performed by PDC and BRM. For more information, see "About PDC REST Services Manager" in *PDC REST Services Manager Integration Guide*.

To configure PDC REST Services Manager in BRM cloud native:

1. Override the PDC REST Services Manager-specific keys in the **values.yaml** file. See "Adding PDC REST Services Manager Keys".
2. Configure OAuth authentication:
   a. If you are using Oracle Access Management for OAuth, create an identity domain, resource server, and OAuth client for PDC REST Services Manager in Oracle Access

Copyright © 2019, 2026, Oracle and/or its affiliates.

Management as described in "Setting Up OAuth for PDC REST Services Manager with Oracle Access Management" in *BRM Security Guide*.

b. Configure the keys in the **override-values.yaml** file for OAuth with either Oracle Identity Cloud Service or Oracle Access Management as described in "Configuring OAuth Authentication in PDC REST Services Manager".

3. Configure outbound communication to the enterprise product catalog. See "Configuring Requests to the Enterprise Product Catalog".
4. Enable TLS encryption in PDC REST Services Manager to secure the communications it receives from your enterprise product catalog. See "Enabling TLS in PDC REST Services Manager".
5. Enable the T3S protocol in PDC REST Services Manager to secure its communications to PDC. See "Enabling T3S in PDC REST Services Manager".
6. Map TMF620 priceType values to BRM events to ensure that PDC REST Services Manager triggers the correct charging events for your pricing components. See "Configuring Mapping of TMF620 priceType to BRM Events".

Adding PDC REST Services Manager Keys

Table 8-6 lists the keys that directly impact PDC REST Services Manager. Add these keys to your **override-values.yaml** file with the same path hierarchy.

> ⚠ **Caution**
>
> Keys with the path **ocpdcrsm.secretValue** hold sensitive data. Handle them carefully with controlled access to the override file containing their values. Encode all of these values in Base64. See "Secrets" in *Kubernetes Concepts*.

Table 8-6 PDC REST Services Manager Keys

Key	Path in Values.yaml File	Description
isEnabled	**ocpdcrsm**	Whether to enable and deploy PDC REST Services Manager with BRM cloud native (**true**) or not (**false**). The default is **false**.
labels.name	**ocpdcrsm**	The string used to form the names of PDC REST Services Manager. The default is **pdcrsm**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-6 (Cont.) PDC REST Services Manager Keys

Key	Path in Values.yaml File	Description
deployment.*	**ocpdcrsm**	The details for deploying PDC REST Services Manager. • **deadlineSeconds**: The maximum time, in seconds, for a deployment to make progress before it is considered failed. The default is **60**. • **revisionHistLimit**: The maximum number of old ReplicaSets for this deployment to retain. The remaining will be garbage-collected in the background. The default is **10**. • **imageName**: The name of the PDC REST Services Manager image, such as **oracle/pdcrsm**. • **imageTag**: The tag associated with the image, such as **:16.26.0.0.0**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**. • **rootLogLevel**: The root log level. The default is **INFO**. • **appLogLevel**: The application log level. The default is **INFO**. • **JAVA_OPTS**: The Java options to configure. • **JAVA_MEM_OPTS**: The Java memory options to configure.
resources.*	**ocpdcrsm.deployment**	The minimum and maximum CPU and memory resources that containers can use. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
configEnv.*	**ocpdcrsm**	The configuration details for the PDC REST Services Manager. • **name**: The name of the PDC REST Services Manager API ConfigMap. The default is **pdcrsm-configmap-env**. • **rsmListenerPort**: The HTTPS port number assigned to listen for API requests from the enterprise product catalog. The default is **8080**. • **baseUrl**: The base URL with resource details to return in the response of PDC REST Services Manager requests. **Note:** After deployment, you can update this value by editing your **override-values.yaml** file and then doing a Helm upgrade. • **useT3s**: Whether to use T3s for the connection to PDC (**true**) or not (**false**). The default is **true**. • **securityEnabled**: Whether to enable token-based authentication for PDC REST Services Manager (**true**) or not (**false**). • **securityType**: Which OAuth provider to use for token-based authentication. Set this to **oam** for Oracle Access Management or **idcs** for Oracle Identity Cloud Service.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-6 (Cont.) PDC REST Services Manager Keys

Key	Path in Values.yaml File	Description
idcs.*	**ocpdcrsm.configEnv**	The Oracle Identity Cloud Service (IDCS) authentication details. • **inboundOauthUri**: The PDC REST Services Manager inbound OAuth base URI. • **inboundOauthClientId**: The PDC REST Services Manager client ID for inbound OAuth. • **inboundOauthFrontendUri**: The front end URI for inbound OAuth. • **inboundOauthAudience**: The primary audience for inbound OAuth. • **inboundOauthProxyHost**: The proxy host for inbound OAuth, if required. • **inboundOauthPubEventScope**: The scope required to access the TMF 620 Publish Event endpoint for inbound OAuth. • **inboundOauthMetricsScope**: The scope required to access the metrics endpoint for inbound OAuth.
oam.*	**ocpdcrsm.configEnv**	The Oracle Access Manager authentication details. • **domainName**: The Oracle Access Manager domain name. • **audience**: The name of the Oracle Access Manager OAuth server. • **endpointURL**: The OAuth token endpoint for Oracle Access Manager. • **introspectendpointuri**: The introspect endpoint for Oracle Access Manager, such as **http://***oam_hostname***:***oam_port***/oauth2/rest/token/introspect**. • **scopeaudience**: The OAuth scope audience for Oracle Access Manager. • **authorizationendpointuri**: The OAuth authorization endpoint for Oracle Access Manager. • **proxyhost**: The OAuth proxy host for Oracle Access Manager. • **frontenduri**: The OAuth front end URI for Oracle Access Manager.
isTracingEnabled	**ocpdcrsm.configEnv**	Whether to enable tracing for the PDC REST Services Manager API (**true**) or not (**false**). The default is **false**.
isTlsEnabled	**ocpdcrsm.configEnv**	Whether to enable TLS encryption for PDC REST Services Manager (**true**) or not (**false**). The default is **false**.
tlsVersions	**ocpdcrsm.configEnv**	The list of supported TLS versions, such as **TLSv1.2,TLSv1.3**.
tlsCertificateFile	**ocpdcrsm.configEnv**	The path to the TLS certificate bundle relative to the Helm chart. The certificate must be in PKCS12 format. Ensure that the certificate file in the **rsm** directory.
extPDCRSMTlsSecret	**ocpdcrsm.configEnv**	The name of the pre-created Kubernetes Secret for the custom TLS certificate. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 8-6 (Cont.) PDC REST Services Manager Keys

Key	Path in Values.yaml File	Description
httpClients.*	**ocpdcrsm.configEnv**	The details for configuring the HTTP client.
nodeSelector	**ocpdcrsm.configEnv**	The rules for deploying the pod on specific nodes.
affinity	**ocpdcrsm.configEnv**	The rules for running the pod on specific nodes. Set this key if you want to constrain the pod to run only on the nodes that meet your criteria. For more information about this key, see "Node Affinity" in the Kubernetes documentation.
addOnPodSpec	**ocpdcrsm.configEnv**	The details for extending pod specification or overriding features. By default, this key is empty.
monitoring.prometheus*	**ocpdcrsm.configEnv**	The details for monitoring PDC REST Services Manager. • **isEnabled**: Whether you are using Prometheus Operator to monitor PDC REST Services Manager (**true**) or not (**false**). The default is **false**. • **namespace**: The namespace in which to deploy Prometheus Operator. The default is **monitoring**.
secretValue.*	**ocpdcrsm.rsm**	The credentials for accessing the system. • **name**: The name of the Kubernetes Secret that copies certificates to the container. The default is **pdcrsm-secret-env**. • **inboundOauthClientSecret**: The client secret for PDC REST Services Manager inbound OAuth. • **tlsCertificatePassphrase**: The Base64-encoded passphrase for the TLS certificate. • **httpClients.***: The HTTP client configuration.
extPdcrsmSecret	**ocpdcrsm.secret**	The name of the external Kubernetes Secret for the PDC REST Services Manager. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
service.*	**ocpdcrsm**	Details about the pdcrsm service: • **name**: The service name: **pdcrsm**. • **type**: The service type. The default is **ClusterIP**. • **nodePort**: The external node port. The default is **31000**. **Note**: The **nodePort** key is applicable only when the **type** key is set to **NodePort**.

Sample PDC REST Services Manager override-values.yaml Entries

The following shows sample content in the **override-values.yaml** for PDC REST Services Manager, when Oracle Access Management is used for OAuth authentication:

```
ocpdcrsm:
    isEnabled: true
    labels:
        name: "pdcrsm"
    deployment:
        deadlineSeconds: 60
        revisionHistLimit: 10
        imageName: "oracle/pdcrsm"
        imageTag: ":16.26.0.0.0"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    imagePullPolicy: IfNotPresent
    rootLogLevel: INFO
    appLogLevel: INFO
configEnv:
    name: "pdcrsm-configmap-env"
    rsmListenerPort: 8080
    baseURL: xxxxx.xxx.xxxxx.xxx
    useT3s: true
    securityEnabled: true
    securityType: oam
    resources:
        requests:
           cpu: "50m"
           memory: "256Mi"
        limits:
           cpu: "1000m"
           memory: "2Gi"
    oam:
        domainName: PDCRSMDomain
        audience: PDCRSMResourceServer
        endpointURL: http://oam_host:oam_port/oauth2/rest/token
        introspectendpointuri: http://oam_host:oam_port/oauth2/rest/token/introspect
        scopeaudience: http://oam_host:oam_port/
        authorizationendpointuri: http://oam_host:oam_port/oauth2/authorize
        proxyhost: http://proxyhost:proxyport/
        frontenduri: http://oam_host:oam_port
secretValue:
    name: "pdcrsm-secret-env"
service:
    name: "pdcrsm"
    type: "ClusterIP"
```

Configuring OAuth Authentication in PDC REST Services Manager

PDC REST Services Manager uses the OAuth 2.0 protocol to authenticate an enterprise product catalog's identity and to authorize the enterprise product catalog to access the PDC REST Services Manager API. It does this by validating an OAuth access token that is passed in the header of every HTTP/HTTPS request to the PDC REST Services Manager API.

To configure OAuth authentication in PDC REST Services Manager:

1. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart**:
   - If you are using Oracle Identity Cloud Service (IDCS) for OAuth:
     - **ocpdcrsm.configEnv.isInboundOauthEnabled**: Set this to **true** to enable OAuth authentication.
     - **ocpdcrsm.configEnv.inboundOauthUri**: Set this to the base URL of your Oracle Identity Cloud Service (IDCS) instance in this format:

       `https://idcs-TenantID.identity.oraclecloud.com`
     - **ocpdcrsm.configEnv.inboundOauthClientId**: Set this to the client ID of your confidential application.
     - **ocpdcrsm.secretValue.inboundOauthClientSecret**: Set this to the Base64-encrypted client secret obtained from your IDCS application.
     - **ocpdcrsm.configEnv.inboundOauthFrontendUri**: Set this to the base URL of your confidential application when run, such as http://myapp.example.com:8080.

Copyright © 2019, 2026, Oracle and/or its affiliates.

  - **ocpdcrsm.configEnv.inboundOauthAudience**: Set this to the primary audience as provisioned for the PDC REST Services Manager application in IDCS.
  - **ocpdcrsm.configEnv.inboundOauthProxyHost**: Set this to the host name of your proxy server, if required.
  - **ocpdcrsm.configEnv.inboundOauthPubEventScope**: Set this to the name of the scope for accessing the TMF620 Publish Event endpoint for inbound OAuth authentication, such as **pubevent**.
  - **ocpdcrsm.configEnv.inboundOauthMetricsScope**: Set this to the name of the scope for accessing the metrics endpoint for inbound OAuth authentication, such as **metrics**.

- If you are using Oracle Access Management for OAuth:
  - **ocpdcrsm.configEnv.oam.domainName**: Set this to the name of the OAuth identity domain created in Oracle Access Management for PDC REST Services Manager.
  - **ocpdcrsm.configEnv.oam.audience**: Set this to the name of the OAuth resource server created in Oracle Access Management for PDC REST Services Manager.
  - **ocpdcrsm.configEnv.oam.endpointURL**: Set this to the URL for requesting an OAuth token from Oracle Access Management.
  - **ocpdcrsm.configEnv.oam.introspectendpointuri**: Set this to the URL for validating an OAuth token from Oracle Access Management, such as **http://***oam_hostname***:***oam_port***/oauth2/rest/token/introspect**.
  - **ocpdcrsm.configEnv.oam.scopeaudience**: Set this to the primary audience for PDC REST Services Manager in the Oracle Access Management resource, used for error handling. This is the same as **ocpdcrsm.configEnv.oam.frontenduri**, ending with **/**.
  - **ocpdcrsm.configEnv.oam.authorizationendpointuri**: The URL for authorizing role-based access. PDC REST Services Manager does not support role-based access, so this will not be used.
  - **ocpdcrsm.configEnv.oam.proxyhost**: Set this to the URL for your Oracle Access Management proxy server, if needed.
  - **ocpdcrsm.configEnv.oam.frontenduri**: Set this to the URL for of the OAuth client created in Oracle Access Management for PDC REST Services Manager.

2. Run the **helm upgrade** command to update the Helm release:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
   ```

3. Restart the PDC REST Services Manager pods. If downtime is not a concern, both pods can be deleted and re-created by running the following command. Otherwise, delete one pod at a time, waiting for its replacement pod to become "Running" before deleting the next one.

   ```
   kubectl -n BrmNameSpace delete pods --selector=app.kubernetes.io/name=pdcrsm
   ```

Configuring Requests to the Enterprise Product Catalog

PDC REST Services Manager sends requests to the enterprise product catalog when calling the enterprise product catalog's REST API and when publishing acknowledgment notifications.

To configure PDC REST Services Manager to send requests to the enterprise product catalog:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Open the **override-values.yaml** file for **oc-cn-helm-chart**.
2. Edit the keys in the file based on the type of authentication required by your enterprise product catalog:
   - For OAuth 2.0 authentication, edit the keys in Table 8-7.

Table 8-7 OAuth 2.0 Keys

Key	Path in Values.yaml file	Description
tokenEndpoint	**ocpdcrsm.configEnv.httpClients.security.oauth2**	The endpoint used to retrieve a token from.
clientId	**ocpdcrsm.configEnv.httpClients.security.oauth2**	The client ID used to authenticate the request from PDC REST Services Manager.
scope	**ocpdcrsm.configEnv.httpClients.security.oauth2**	The scopes required by the enterprise product catalog.
grantType	**ocpdcrsm.configEnv.httpClients.security.oauth2**	The grant type to be used for the OAuth flow: **client_credentials** or **password**.
clientSecret	**ocpdcrsm.secretValue.httpClients.security.oauth2**	The encrypted client secret used to authenticate the request from PDC REST Services Manager.
password	**ocpdcrsm.secretValue.httpClients.security.oauth2**	The encrypted password required for accessing the enterprise product catalog.

   - For basic authentication, edit the keys in Table 8-8.

Table 8-8 basicAuth Keys

Key	Path in Values.yaml file	Description
username	**ocpdcrsm.configEnv.httpClients.security.basicAuth**	The user name required for accessing the enterprise product catalog.
password	**ocpdcrsm.secretValue.httpClients.security.basicAuth**	The password required for accessing the enterprise product catalog.

3. Run the **helm upgrade** command to update the Helm release:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
   ```

4. Restart the PDC REST Services Manager pods. If downtime is not a concern, both pods can be deleted and re-created by running the following command. Otherwise, delete one pod at a time, waiting for its replacement pod to have a "Running" status before deleting the next one.

   ```
   kubectl --namespace BrmNameSpace delete pods --selector=app.kubernetes.io/name=pdcrsm
   ```

The following shows an example configuration for OAuth 2.0 authentication.

> **Note**
>
> All **urlRegex** values in the file must be properly escaped with ****. The characters that must be escaped are: **\.[]{}()<>*+-=!? ^$|**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
configEnv:
    httpClients:
        - urlRegex: "http://hostname:port/mobile/custom/catalogManagement/.*"
          security:
              oauth2:
                  tokenEndpoint: "https://idcs_hostname/oauth2/v1/token"
                  clientId: "fcb3443f6c504ed789ba38a78341b88a"
                  scope: "https://hostnameurn:opc:resource:consumer::all"
                  grantType: EncryptedPassword
secretValue:
    httpClients:
        - urlRegex: "http://hostname:port/mobile/custom/catalogManagement/.*"
          security:
              oauth2:
                  clientSecret: client_secret
                  password: EncryptedPassword
```

The following shows an example configuration for Basic authentication:

> **Note**
>
> All **urlRegex** values in the file must be properly escaped with ****. The characters that must be escaped are: **\.[]{}()<>*+-=!? ^$|**.

```
configEnv:
    httpClients:
        - urlRegex: "http://hostname:port/mobile/custom/PublishingAPI.*"
          security:
              basicAuth:
                 username: eccUser
secretValue:
    httpClients:
        - urlRegex: "http://hostname:port/mobile/custom/PublishingAPI.*"
          security:
              basicAuth:
                  password: EncryptedPassword
```

Enabling TLS in PDC REST Services Manager

You can enable TLS encryption in PDC REST Services Manager to secure the communications it receives from your enterprise product catalog.

To enable TLS in PDC REST Services Manager:

1. Generate a self-signed SSL certificate:
   a. Create a directory for storing your SSL certificate that is accessible by the BRM Helm chart, such as **oc-cn-helm-chart/rsm_cert**.
   b. Generate an SSL certificate. For example, this creates a certificate file named **cert.pem**:

      ```
      openssl req -x509 -newkey rsa:4096 -keyout openSSLKey.pem -out cert.pem -days 365 -nodes
      ```

   c. Generate a PKCS12 KeyStore file. For example, this creates a KeyStore file named **keystore.p12**:

      ```
      openssl pkcs12 -export -out keyStore.p12 -inkey openSSLKey.pem -in cert.pem
      ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart**:
   - **ocpdcrsm.configEnv.isTlsEnabled**: Set this to **true** to enable TLS encryption for PDC REST Services Manager.
   - **ocpdcrsm.configEnv.tlsVersions**: Set this to the list of supported TLS versions, such as **TLSv1.2,TLSv1.3**.
   - **ocpdcrsm.configEnv.tlsCertificateFile**: Set this to the path to the TLS certificate bundle in the Helm chart. The certificate must be in PKCS12 format.
   - **ocpdcrsm.secretValue.tlsCertificatePassphrase**: Set this to the Base64-encrypted passphrase for the TLS certificate.
3. Run the **helm upgrade** command to update the Helm release:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
   ```
4. To apply the changes, re-create any previously existing PDC REST Services Manager pods:

   ```
   kubectl --namespace BrmNameSpace delete pods --selector=app.kubernetes.io/name=pdcrsm
   ```

After you enable TLS, connect to PDC REST Services Manager services using HTTPS only.

Enabling T3S in PDC REST Services Manager

Enable the T3S protocol in PDC REST Services Manager to secure its communications to PDC.

To enable T3S in PDC REST Services Manager:

1. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart**:
   - **ocpdcrsm.configEnv.useT3s**: Set this to **true**.
2. Run the **helm upgrade** command to update the Helm release:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
   ```
3. To apply the changes, re-create any previously existing PDC REST Services Manager pods:

   ```
   kubectl --namespace BrmNameSpace delete pods --selector=app.kubernetes.io/name=pdcrsm
   ```

Configuring Mapping of TMF620 priceType to BRM Events

If you are using PDC REST Services Manager, you must configure the mappings of BRM event names to the values your enterprise product catalog sends in the **priceType** property of the **ProductOfferingPrice** element of the TMF 620 payload.

The mappings are configured in **configmap_pdcrsm_appeventCfg.yaml**. You can add mappings as needed for your deployment or use the default mappings provided at installation.

To add or edit mappings:

1. Open the **configmap_pdcrsm_appeventCfg.yaml** file.
2. Edit the existing mappings, or use them as templates to add new ones. Use the following format:

   ```
   pricetype : "eventname"
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

where:

- *pricetype* is the value sent in the **priceType** property of the **ProductOfferingPrice** element of the TMF620 payload.
- *eventname* is the name of the BRM event the price type should be mapped to.

For example, the default mappings for one-time fees and usage events are:

```
ONE_TIME : "EventBillingProductFeePurchase"
ONE_TIME_PRICE_PLAN : "EventBillingProductFeePurchase"
USAGE : "EventSession"
USAGE_PRICE_PLAN : "EventSession"
```

3. Run the **helm upgrade** command to update the Helm release:

   helm upgrade *BrmReleaseName* **oc-cn-helm-chart --values** *OverrideValuesFile* **--namespace** *BrmNameSpace*

4. Restart the PDC REST Services Manager pods. If downtime is not a concern, both pods can be deleted and re-created by running the following command. Otherwise, delete one pod at a time, waiting for its replacement pod to become "Running" before deleting the next one.

   kubectl --namespace *BrmNameSpace* **delete pods --selector=app.kubernetes.io/name=pdcrsm**

Copyright © 2019, 2026, Oracle and/or its affiliates.

9

Configuring the BRM Client Services

Learn how to configure Billing Care, Billing Care REST API, Collections Configuration Center, and Business Operations Center to run in your Oracle Communications Billing and Revenue Management (BRM) cloud native environment.

Topics in this document:

- About Configuring Your BRM Client Services
- Configuring Business Operations Center
- Configuring Collections Configuration Center
- Configuring Billing Care
- Configuring the Billing Care REST API

About Configuring Your BRM Client Services

Business Operations Center, Billing Care, and Billing Care REST API share a similar image stack.

Figure 9-1 shows the process for deploying Billing Care using WebLogic Operator. The same process is used for the Billing Care REST API. The only difference is the name of the deployer: **bcws-domain-deployer**.

Figure 9-1 Billing Care Deployment Flow

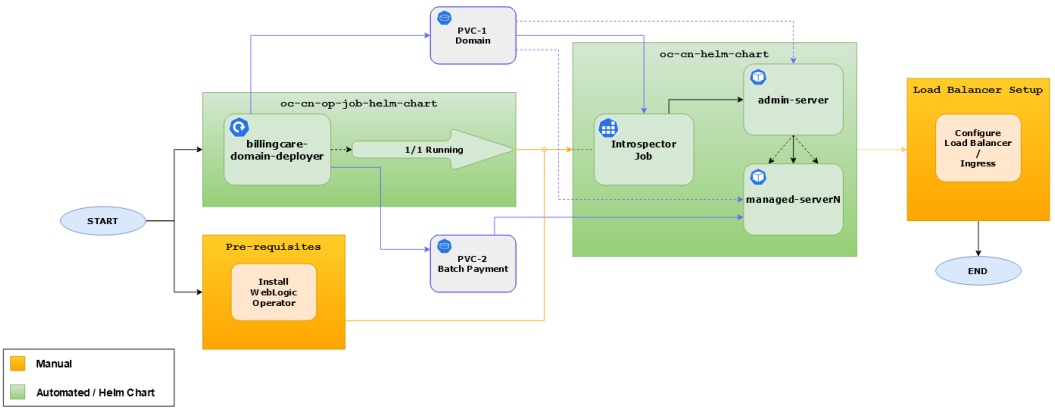
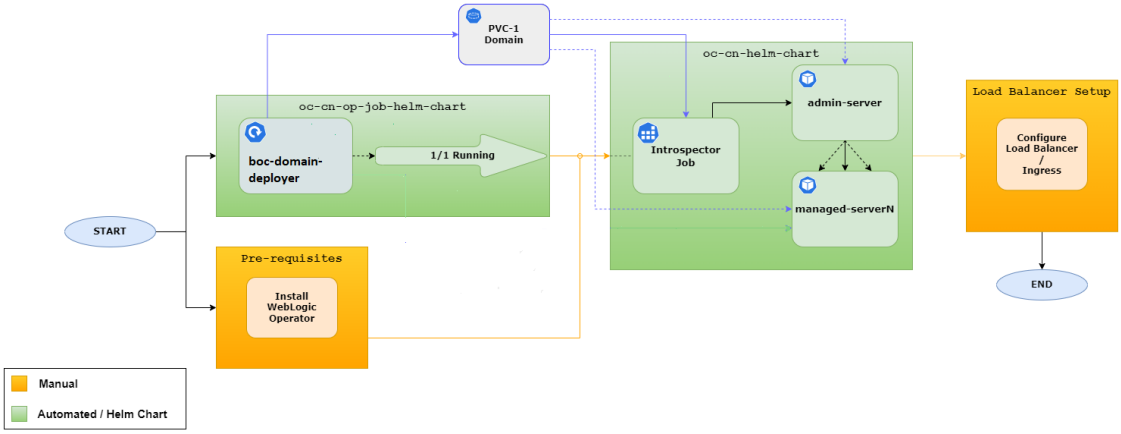


Figure 9-2 shows the process for deploying Business Operations Center using WebLogic Operator. It is similar to the Billing Care process.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 9-2 Business Operations Center Deployment Flow



> **Note**
>
> It is important to wait until the *component*-**domain-deployer** process is in the **1/1 Running** status before running **oc-cn-helm-chart**.

You deploy these services by using the following Helm charts:

- **oc-cn-op-job-helm-chart**: This chart creates and configures the WebLogic domain, deploys the application, deploys and links the SDK (for Billing Care and Billing Care REST API), and loads the authorization policies.
- **oc-cn-helm-chart**: This chart starts the rolling restart of the WebLogic servers and the application update.
- **WebLogic Operator chart**: This chart manages the application domain, controlling the service availability when managed server pods are scaled up or down.

Configuring Business Operations Center

Business Operations Center is a web-based client application that you use to run business operations such as billing, invoicing, and payment collections. For more information, see "Using Business Operations Center" in *BRM System Administrator's Guide*.

To configure Business Operations Center to run in your BRM cloud native environment:

1. Override the Business Operations Center-specific keys in the **values.yaml** file for **oc-cn-op-job-helm-chart**. See "Adding Business Operations Center Keys for oc-cn-op-job-helm-chart".
2. Override the Business Operations Center-specific keys in the **values.yaml** file for **oc-cn-helm-chart**. See "Adding Business Operations Center Keys for oc-cn-helm-chart".
3. Set up volume mounts. See "About Business Operations Center Volume Mounts".
4. Create a WebLogic domain and install the Business Operations Center application. See "Creating a WebLogic Domain and Installing the Business Operations Center Application".
5. Set up SAML for SSO in Business Operations Center. See "Setting Up SSO for Business Operations Center".

Copyright © 2019, 2026, Oracle and/or its affiliates.

6. Set up local users and groups for Business Operations Center. See "Setting Up Local Users and Groups for Business Operations Center".
7. Start and stop your WebLogic servers. See "Starting and Stopping WebLogic Servers".

> **Note**
>
> To set up Business Operations Center, ensure that you successfully complete the installation of **oc-cn-op-job-helm-chart** before you install or upgrade **oc-cn-helm-chart**.

Adding Business Operations Center Keys for oc-cn-op-job-helm-chart

Table 9-1 lists the keys that directly impact Business Operations Center. Add these keys to your **override-values.yaml** file for **oc-cn-op-job-helm-chart** with the same path hierarchy.

For a complete set of keys to personalize Business Operations Center deployment, see the keys with the path **ocboc.boc** in the **oc-cn-op-job-helm-chart/values.yaml** file.

> **Caution**
>
> Keys with the path **ocboc.boc.secretVal** hold sensitive data. Handle them carefully with controlled access to the file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Table 9-1 Keys for oc-cn-op-job-helm-chart

Key	Path in Values.yaml file	Description
isEnabled	**ocboc.boc**	Whether to deploy, configure, and start Business Operation Center services. • **false**: Kubernetes resources meant for the Business Operation Center application will not be created. • **true**: Creates the necessary Kubernetes resources for using Business Operation Center. This is the default.
deployment.*	**ocboc.boc**	The details for deploying the Business Operations Center pod.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-1 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in Values.yaml file	Description
configEnv.*	**ocboc.boc**	• **managedHttpPort**: The container port for access to the managed server. The default is **8001**. • **httpPort**: The container port for access to the WebLogic domain over HTTP. The default is **7011**. • **serverStartMode**: The mode in which to start the server: **dev** or **prod**. The default is **prod**. • **adminUser**: The name of the user who is granted administrator rights to the WebLogic domain. The default is **weblogic**. • **dbhost**: The host name or IP address for the database server. • **dbPort**: The port number for the database server. • **dbServiceName**: The service name for the database. • **dbSSLMode**: The SSL mode: **NO**, **YES**, or **ONE_WAY**. • **dbWalletType**: The database SSL wallet type, such as SSO. • **extDBSSLWalletSecret**: The name of the external Kubernetes Secret containing the SSL database wallet. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **rcuSysDBAUser**: The database administrator user name. • **rcuDBARole**: The role of the database administrator user. • **rcuPrefix**: The prefix for the OPSS schema. The default is **BOC11**. • **rcuCreate**: Whether to drop the existing OPSS schema (**true**) or not (**false**). The default is **true**. • **rcuArgs**: The additional arguments for creating the RCU. • **rcuTablespace**: Whether to drop the existing OPSS schema (**true**) or not (**false**). The default is **true**. • **rcuTempTablespace**: The name of an existing temporary tablespace in your database. If left empty, new tablespaces are created with names starting with **rcuPrefix**. • **isOPSS**: Whether to create an OPSS domain (**true**) or a non-OPSS domain (**false**). The default is **true**. • **extAccessPolicyCM**: The name of the ConfigMap containing the policy file. • **isLDAPEnabled**: Whether to skip creation of the Oracle Unified Directory Authenticator (**true**) or not (**false**). The default is **true**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-1 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in Values.yaml file	Description
		• **ldapAdmin**: The Distinguished Name to connect to the LDAP server. The default is **cn=Directory Manager**. • **ldapHost**: The host name or IP address of the LDAP Server (for example, OUD) where users and groups are configured for access to Business Operations Center. • **ldapPort**: The port number on which the LDAP server is listening. The default is **389**. • **ldapGroupBase**: The LDAP base DN that contains groups. • **ldapUserBase**: The LDAP base DN that contains users. • **ldapProviderName**: The name of Authentication Provider. The default is **OUDAuthenticator**. • **bocSchemaUserName**: The Business Operations Center database schema user name. The default is **bocdb**. • **bocSchemaBocTablespace**: The default tablespace for the Business Operations Center database administrator. The default is **boc_default_tbls**. • **bocSchemaTempTablespace**: The temp tablespace for the Business Operations Center database administrator. The default is **boc_temp_tbls**. • **billingCareUrl**: The URL of the Billing Care instance that is used with your BRM Server. Leave this blank if Billing Care isn't installed in your environment. • **logoutURL**: The URL where the user is redirected after logging out from the application. The default is **login.html**. • **timeoutWaringDuration**: The default is **90**. • **pageSize**: The default is **25**. • **refreshInterval**: The default is **28800000**. • **connectionTimeout**: The default is **16000**. • **dbURL**: Used to create the WebLogic data source for connecting to the Business Operations Center schema. This is also the connection string for the database where schemas needed by Oracle Fusion Middleware products are created, especially OPSS. Use one of these formats: – *DatabaseHost***:***DatabasePort***/***ServiceName* – *DatabaseHost***:***DatabasePort***:***ServiceID* • **enablePvt**: Whether to consider **pin_virtual_time** when running business operations jobs. The default is **true**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-1 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in Values.yaml file	Description
		• **targetServer**: The name of the cluster server. The default is **cluster-1**. • **keystoreAlias**: The private key alias of the KeyStore. • **extKeystoreSecret**: The names of the pre-created Kubernetes Secrets for the Business Operations Center KeyStore certificates and wallets. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **keystoreType**: The file type of the SSL Identity and Trust store, which is either **PKCS12** or **JKS**. The default is **PKCS12**. • **keystoreIdentityFileName**: The file name of the Identity KeyStore. • **keystoreTrustFileName**: The file name of the Trust KeyStore. • **dbSSLMode**: The type of connection required to connect to the database: **Yes-Two Way**, **Yes-One Way**, or **No**. • **dbWalletType**: The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**. • **tlsVersions**: The list of TLS versions to support for connection with the WebLogic domain. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**. • **isSSOEnabled**: Whether to enable single sign-on (SSO) for Business Operations Center cloud native services using SAML 2.0 (**true**) or not (**false**) The default is **false**. • **extMetadataCM**: The name of the external ConfigMap containing the IDP metadata file. • **samlAsserterName**: The name of the SAML Asserter. The default is **samlBOCAsserter**. • **ssoPublishedSiteURL**: The base URL that is used to construct endpoint URLs. This is typically the Load Balancer host and port at which the server is visible externally. It must be appended with **/saml2**. For example: **https://***LoadBalancerHost***:***LoadBalancerPort***/saml2**. • **ssoDefaultURL**: The URL where unsolicited authentication responses are sent if they do not contain an accompanying target URL. • **reloadVersion**: Update this value with any value different from the current value to force a restart of the deployer. The default is **1**. • **reset**: Whether to wipe all previous states and do a fresh setup of the domain. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-1 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in Values.yaml file	Description
secretVal.*	**ocboc.boc**	• **adminPassword**: The Base64-encoded password for the WebLogic domain's administrative user. This is used for accessing the WebLogic Remote Console for administrative operations. • **ldapPassword**: The Base64-encoded password of the LDAP Server admin user. • **rcuSysDBAPassword**: The Base64-encoded database administrator's password. • **rcuSchemaPassword**: The Base64-encoded password for schemas of Oracle Fusion Middleware products that will be created by RCU, which is used by OPSS. • **bocSchemaPassword**: The Base64-encoded Business Operations Center database schema password. • **dbWalletPassword**: The password for accessing the certificates from the TrustStore and KeyStore. • **keystoreIdentityPassword**: The StorePass for the Identity KeyStore. • **keystoreKeyPassword**: The KeyPass for the Identity KeyStore. • **keystoreTrustPassword**: The StorePass for the Trust KeyStore.
secret.*	**ocboc.boc**	The details about the Business Operations Center external Kubernetes Secrets. • **extFMWUserSecret**: The name of the external Kubernetes Secret containing the Fusion Middleware user passwords. • **extSetupSecret**: The name of the external Kubernetes Secret containing the set up passwords. • **extWLSUsersSecret**: The name of the external Kubernetes Secret containing the WebLogic Server user passwords. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-1 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in Values.yaml file	Description
wop.*	**ocboc.boc**	The details for configuration the WebLogic Server. • **domainUID**: The name of the domain. The default is **boc-domain**. • **domainRootDir**: The location within the container where the domain is created. The default is **/shared**. • **totalManagedServers**: The number of managed servers in the cluster. The default is **5**. • **initialServerCount**: The number of managed servers initially started for the domain. The default is **2**. • **adminChannelPort**: The NodePort where the admin-server's HTTP service will be accessible. The default is **30811**. **Note:** Set this key only if you want the **boc-domain-admin-server-ext** service to deploy as NodePort. • **serverStartPolicy**: The WebLogic servers that the Operator starts when it discovers the domain: **NEVER**, **ADMIN_ONLY**, or **IF_NEEDED**. The default is **IF_NEEDED**.
resources.*	**ocboc.boc**	The minimum and maximum CPU and memory resources for the cm pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
volume.domain.*	**ocbc.boc**	Details about the PVC for the domain file system: • **storage**: The storage size of the volume. • **createOption**: By default, the boc pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.
wlsUserGroups.*	**ocboc.boc**	The details for setting up the users and groups to the domain's DefaultAuthenticator.
scriptsConfigName	**ocboc.boc.extensions**	The name of the ConfigMap containing the scripts for running additional steps to configure the domain or application.
nodeSelector	**ocboc.boc**	The rules for scheduling pods on particular nodes using simple selectors using Node Selector rules.
affinity	**ocboc.boc**	The rules for scheduling pods on particular nodes using more powerful selectors using affinity rules.
addOnPodSpec	**ocboc.boc**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Adding Business Operations Center Keys for oc-cn-helm-chart

Table 9-2 lists the keys that directly impact Business Operations Center. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart** with the same path hierarchy.

For a complete set of keys to personalize Business Operations Center deployment, see the keys with the path **ocboc.boc** in the **oc-cn-helm-chart/values.yaml** file.

> ⚠ **Caution**
>
> Keys with the path **ocboc.boc.secretVal** hold sensitive data. Handle them carefully with controlled access to the file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Table 9-2 Keys for oc-cn-helm-chart

Key	Path in Values.yaml file	Description
isEnabled	**ocboc.boc**	Whether to deploy, configure, and start Business Operation Center services. • **false**: Kubernetes resources meant for the Business Operation Center application will not be created. • **true**: Creates the necessary Kubernetes resources for using Business Operation Center. This is the default.
deployment.*	**ocboc.boc**	The details for deploying the Business Operations Center pod.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-2 (Cont.) Keys for oc-cn-helm-chart

Key	Path in Values.yaml file	Description
configEnv.*	**ocboc.boc**	The details for configuring Business Operations Center. • **dbhost**: The host name or IP address for the database server. • **dbPort**: The port number for the database server. • **dbServiceName**: The service name for the database. • **dbSSLMode**: The SSL mode: **NO**, **YES**, or **ONE_WAY**. • **dbWalletType**: The database SSL wallet type, such as SSO. • **extDBSSLWalletSecret**: The name of the external Kubernetes Secret containing the SSL database wallet. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **httpPort**: The container's port for accessing the WebLogic domain over HTTP. The default is **7011**. • **isOPSS**: Whether to create an OPSS domain (**true**) or a non-OPSS domain (**false**). Set this to **true** for production systems. The default is **true**. • **keystoreAlias**: The private key alias for the KeyStore. • **extKeystoreSecret**: The name of the external Kubernetes Secret containing Identity and Trust KeyStore files. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **keystoreType**: The file type of the SSL Identity and Trust KeyStore: **PKCS12** or **JKS**. The default is **PKCS12**. • **runUpgrade**: Whether to run the database schema upgrade. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-2 (Cont.) Keys for oc-cn-helm-chart

Key	Path in Values.yaml file	Description
infranet.*	**ocboc.boc**	The details for configuring Business Operations Center. • **user.login**: The user name of the service with permission to access BRM, such as **boc_client.0.0.0.1**. • **user.serviceType**: The POID type of the service that has permission to access BRM. The default is **/service/admin_client**. • **user.serviceID**: The POID ID of the service that has permission to access BRM. The default is **415**. • **connectionpool.minSize**: The minimum number of connections allowed in the pool. The default is **25**. • **connectionpool.maxSize**: The maximum number of connections allowed in the pool. The default is **50**. • **loglevel**: The log level for the **Infranet.properties** file. The default is **3**. • **addOnProperties**: This field is empty by default. You can use this key to specify custom **Infranet.properties** values.
secretVal.*	**ocboc.boc**	The password details. • **walletPassword**: The Business Operations Center wallet password. This value must be Base64-encoded. • **keystoreTrustPassword**: The StorePass of the Trust Keystore, which is used for setting up the SSL-enabled domain. This value must be Base64-encoded.
secret.*	**ocboc.boc**	The details about the Business Operations Center external Kubernetes Secrets. • **extAppSecret**: The name of the external Kubernetes Secret containing the application passwords. • **extFMWUserSecret**: The name of the external Kubernetes Secret containing the Fusion Middleware user passwords. • **extSetupSecret**: The name of the external Kubernetes Secret containing the set up passwords. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-2 (Cont.) Keys for oc-cn-helm-chart

Key	Path in Values.yaml file	Description
wop.*	**ocboc.boc**	The details for configuration the WebLogic Server. • **domainUID**: The name of the domain. The default is **boc-domain**. • **domainRootDir**: The location within the container where the domain is created. The default is **/shared**. • **totalManagedServers**: The number of managed servers in the cluster. The default is **5**. • **initialServerCount**: The number of managed servers initially started for the domain. The default is **2**. • **adminChannelPort**: The NodePort where the admin-server's http service will be accessible. The default is empty. **Note:** Set this key only if you want the **boc-domain-admin-server-ext** service to deploy as NodePort. • **serverStartPolicy**: The WebLogic servers that the Operator starts when it discovers the domain: **NEVER**, **ADMIN_ONLY**, or **IF_NEEDED**. The default is **IF_NEEDED**. • **restartVersion**: Whether to force a rolling restart of all server pods. To force the restart, set it to any value other than **1**. The default is **1**. • **introspectVersion**: Whether to force a domain introspection. To do so, set it to any value other than **1**. The default is **1**.
monitoring.*	**ocboc.boc**	The details for monitoring and autoscaling Business Operations Center. By default, monitoring is disabled. See "Monitoring and Autoscaling Business Operations Center Cloud Native" in *BRM Cloud Native System Administrator's Guide*.
scriptsConfigName	**ocboc.boc.extensions**	The name of the ConfigMap containing the scripts for running additional steps to configure the domain or application.
resources.*	**ocboc.boc**	The minimum and maximum CPU and memory resources for the cm pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
nodeSelector	**ocboc.boc**	The rules for scheduling pods on particular nodes using simple selectors using Node Selector rules.
affinity	**ocboc.boc**	The rules for scheduling pods on particular nodes using more powerful selectors using affinity rules.
addOnPodSpec	**ocboc.boc**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Updating Infranet.properties for Business Operations Center

The **Infranet.properties** file entries are located in the **values.yaml** file. This makes it easier to update them.

Following is a sample configuration block (located in the **ocboc.boc** path in **oc-cn-helm-chart**) for the **Infranet.properties** entries:

```
infranet:
    user:
        login: 'boc_client.0.0.0.1'
        serviceType: '/service/admin_client'
        serviceId: 2
    connectionpool:
        minSize: 25
        maxSize: 50
    logLevel: 3
    addOnProperties: ""
```

If you have custom properties, they should be defined here using the **addOnProperties** key. For example:

```
addOnProperties: |-
    infranet.connectionpool.timeout=90000
    infranet.pcp.debug.flags=0x3FFF
    infranet.pcp.debug.enabled=true
```

To update these properties, update the values in **oc-cn-helm-chart** and change the value of **ocboc.boc.wop.restartVersion** in **oc-cn-helm-chart** to any new value. This will force a pod restart and the new values will be used.

Adding Custom Configuration to Deployment Workflow for Business Operations Center

You can provide additional configuration to be applied at particular checkpoints in the Business Operations Center deployment workflow. These checkpoints are:

- **ext_deployer_pre_exit**: Called after the standard configuration in **deployer.sh** in **oc-cn-op-job-helm-chart**
- **ext_init_app_pre_exit**: Called after the standard configuration in the init-app **initContainer** container in both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**
- **ext_init_config_pre_exit**: Called after the standard configuration in the init-config **initContainer** container in both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**
- **ext_init_upgrade_pre_exit**: Called after the standard configuration in the **upgrade** container

Create a ConfigMap with your configuration scripts, including a shell script named **run_hooks.sh** that calls your other scripts. For example:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ext-scripts
data:
  run_hooks.sh: |+
    #!/bin/bash
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    echo "executing extension for: $@"
    CURRENT_CHECKPOINT=$1
    if [ "$CURRENT_CHECKPOINT" == "ext_deployer_pre_exit" ] ; then
      sh my_deployer_extension.sh
    fi
  my_deployer_extension.sh: |+
    #!/bin/bash
    echo "executing my_deployer_extension"
...
```

Specify the name of your ConfigMap in the **ocboc.boc.extensions.scriptsConfigName** key in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

About Business Operations Center Volume Mounts

The Business Operations Center container requires Kubernetes volume mounts for sharing the domain and application file system between the WebLogic Cluster servers. Business Operations Center requires a volume for the domain. By default, this is created dynamically, using the provisioner defined in BRM, in the **storage-class** key in **oc-cn-op-job-helm-chart**.

To change the volume type or provider, modify the **ocboc.boc.volume.domain.createOption** key in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

Creating a WebLogic Domain and Installing the Business Operations Center Application

The WebLogic domain is created by a Kubernetes Deployment when **oc-cn-op-job-helm-chart** is installed. The same job also installs the Business Operations Center application and deploys the application WAR file onto the WebLogic Cluster.

The **oc-cn-op-job-helm-chart** chart also:

- Creates a Kubernetes ConfigMap and Secrets, which are used throughout the life-cycle of the WebLogic domain.
- Initializes the **PersistentVolumeClaim** for the domain and application file system as well as third-party libraries.

> **Note**
>
> The **override-values.yaml** file that you use for this chart must include BRM override values.

After you install **oc-cn-op-job-helm-chart**, wait until the Kubernetes deployment has reached the **1/1 Running** status. Then, you can install or upgrade **oc-cn-helm-chart** for Business Operations Center services.

After the deployment is running, don't delete the chart. Its resources will be used for starting and stopping the servers through **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Setting Up SSO for Business Operations Center

SSO allows users to log in to applications using a single user name and password combination. You set up SSO for Business Operations Center cloud native services by using SAML 2.0.

To set up SSO for Business Operations Center:

1. Export the SAML 2.0 metadata XML file from your identity and access management (IAM) system.

   For example, if you are using Oracle Access Management, you can export the file by following the instructions in "Exporting Metadata" in *Oracle Fusion Middleware Administering Oracle Access Management*.

2. Add the metadata XML file to your BRM cloud native deployment by doing one of the following:

   - Rename the metadata XML file to **metadata.xml**, and then move **metadata.xml** to the **oc-cn-op-job-helm-chart/boc/idp** directory. In this case, you must leave the **ocboc.boc.configEnv.extMetadataCM** key for **oc-cn-op-job-helm-chart** empty.
   - Pre-create the IDP metadata ConfigMap for Business Operations Center and set the **ocboc.boc.configEnv.extMetadataCM** key in your **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

     For more information, see "Managing Wallet and KeyStore Certificates" in *BRM Cloud Native System Administrator's Guide*.

3. Configure the KeyStores needed by SAML by doing one of the following:

   - Generate the Identity and Trust KeyStores and then move your files, such as **identity.p12** and **trust.p12**, under the **oc-cn-op-job-helm-chart/boc/keystore** directory. In this case, you must leave the **ocboc.boc.configEnv.extKeystoreSecret** key for **oc-cn-op-job-helm-chart** empty.
   - Pre-create the Kubernetes Secret for the Identity and Trust KeyStore files and set the **ocboc.boc.configEnv.extKeystoreSecret** key in your **override-values.yaml** file for both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**.

     For more information, see "Managing Wallet and KeyStore Certificates" in *BRM Cloud Native System Administrator's Guide*.

4. In your **override-values.yaml** file for **oc-cn-op-job-helm-chart**, set the following keys:

   - **ocboc.boc.configEnv.isSSOEnabled**: Set this to **true**.
   - **ocboc.boc.configEnv.keystoreAlias**: Set this to the private key alias of the KeyStore.
   - **ocboc.boc.configEnv.keystoreType**: Set this to the file type of the SSL Identity and Trust store, which is either **PKCS12** or **JKS**. The default is **PKCS12**.
   - **ocboc.boc.configEnv.keystoreIdentityFileName**: Set this to the name of the Identity KeyStore file.
   - **ocboc.boc.configEnv.keystoreTrustFileName**: Set this to the name of the Trust KeyStore file.
   - **ocboc.boc.configEnv.samlAsserterName**: Set this to the name of the SAML Asserter. The default is **samlBOCAsserter**.
   - **ocboc.boc.configEnv.ssoPublishedSiteURL**: Set this to the base URL that is used to construct endpoint URLs. This is typically the load balancer host and port at which

Copyright © 2019, 2026, Oracle and/or its affiliates.

the server is visible externally. It must be appended with **/saml2**. For example: **https://***LoadBalancerHost***:***LoadBalancerPort***/saml2**.

- **ocboc.boc.configEnv.ssoDefaultURL**: Set this to the URL where unsolicited authentication responses are sent if they do not contain an accompanying target URL.
- **ocboc.boc.secretVal.keystoreIdentityPassword**: Set this to the StorePass for the Identity KeyStore.
- **ocboc.boc.secretVal.keystoreKeyPassword**: Set this to the KeyPass for the Identity KeyStore.
- **ocboc.boc.secretVal.keystoreTrustPassword**: Set this to the StorePass for the Trust KeyStore.

5. Configure your load balancer's rules to send responses to the Business Operations Center WebLogic domain with **/saml2** appended to the URL path.

> **Note**
>
> Add this rule to your existing load balancer rules for routing responses to Business Operations Center (**/opsdashboard**), the host name, and so on.

See "Exposing BRM Services Outside the Kubernetes Cluster".

6. Deploy your Business Operations Center cloud native services by following the instructions in "Deploying BRM Cloud Native Services".
7. After Business Operations Center is deployed, retrieve the **sp-metadata-admin-server.xml** file from the **/shared/domains/***domainUID* directory in your container, where *domainUID* is the name of your Business Operations Center domain specified in the **ocboc.boc.wop.domainUID** key.

   The XML file configures the Web SSO Provider Partner. It contains the partner's KeyStore certificates, SAML assertion details, and the URLs where the SAML Identity Provider redirects to provide access to Business Operations Center.
8. Create a profile for your identity provider partner by loading the **sp-metadata-admin-server.xml** file into your IAM system.

   For example, if you are using Oracle Access Management, you can load the file by following the instructions in "Creating Remote Identity Provider Partners" in *Oracle Fusion Middleware Administering Oracle Access Management*.

Setting Up Local Users and Groups for Business Operations Center

You have the option to customize the values for **oc-cn-op-job-helm-chart** to create users and groups locally in Oracle WebLogic Server. This would be especially useful for test environments where you might not have Identity Providers or LDAPs available. The groups for the admin user for WebLogic Server cannot be modified using this procedure.

Any passwords must be encoded using Base64. You can leave the password blank, but then the user will not be able to log in to the application directly.

Copyright © 2019, 2026, Oracle and/or its affiliates.

To set up local users and groups for Billing Care, define the keys under **ocboc.boc.wlsUserGroups** in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**. For example:

```
ocboc:
    boc:
        wlsUserGroups:
            groups:
            -   name: "GroupA"
                description: "GroupA Description"
            -   name: "GroupB"
                description: "GroupB Description"
            users:
            -   name: csr1
                description: "csr1 description"
                password: "Base64_password"
                groups:
                -   "GroupA"
                -   "GroupB"
            -   name: csr2
                description: "csr2 description"
                password: "Base64_password"
                groups:
                -   "GroupB"
```

Starting and Stopping WebLogic Servers

When you install **oc-cn-op-job-helm-chart**, the default configuration sets up a WebLogic Cluster with five Managed Servers. When you install or upgrade **oc-cn-helm-chart** for the Business Operations Center service, two of the managed servers and one Admin Server are started.

By modifying the **override-values.yaml** file for **oc-cn-helm-chart**, you can control:

- The total number of Managed Servers and the initial server start up by using the **totalManagedServers** and **initialServerCount** keys.
- Whether the servers are started or stopped by using the **serverStartPolicy** key. To start the Admin Servers and the Managed Servers in a Cluster, set the key to **IF_NEEDED**. To stop all servers, set the key to **NEVER**.

> **Note**
>
> The keys in the **override-values.yaml** file should be the same as the ones used in **oc-cn-op-job-helm-chart** for keys that are common in both charts.

Before installing or upgrading **oc-cn-helm-chart** for Business Operations Center, ensure that the brm_apps values are configured correctly. If there is a change in any brm_apps values, use **serverStartPolicy** to restart and have the changes take effect.

After you modify the **override-values.yaml** file, update the Helm release for the changes to take effect:

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace
BrmNameSpace
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

where:

- *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.
- *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.
- *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-helm-chart**.

Adding Authorization for a Custom Job Resource

To grant user access to a custom job resource in Business Operations Center:

1. Add the following entry in **oc-cn-op-job-helm-chart/templates/configmap_boc_domain_properties.yaml**:

```
bocauth-config.properties: |
                           JOB_BOC_ADMIN_HAS_ACCESS_TO_RESOURCE =
category_customName_resource
                           ACTIONS_GRANTED_FOR_category_customName_resource =
View,Create,Modify,Delete,Timeline,History
```

   where *customName* is the name of the custom category that has been created or is yet to be created, displayed in Business Operations Center.

Configuring Collections Configuration Center

Collections Configuration Center is a web-based client application that collections agents use to manage overdue balances from your customers. For more information about using Collections Configuration Center, see *Collections Configuration Center Online Help*.

To configure Collections Configuration Center to run in your BRM cloud native environment, override the Collections Configuration Center-specific keys from the **values.yaml** file for **oc-cn-helm-chart**. See "Adding Billing Care Keys for oc-cn-helm-chart".

Adding Collections Configuration Center Keys for oc-cn-helm-chart

Table 9-3 lists a few important keys that directly impact Collections Configuration Center. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart** with the same path hierarchy.

For the complete set of keys to personalize your Collections Configuration Center deployment, see the keys with the path **occcc.ccc** in the **oc-cn-helm-chart/values.yaml** file.

> ⚠ **Caution**
>
> Keys with the path **occcc.ccc.secretVal** hold sensitive data. Handle them carefully with controlled access to the override file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-3 Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
isEnabled	**occcc.ccc**	Whether to deploy, configure, and start Collections Configuration Center services (**true**) or not (**false**). The default is **true**.
deployment.*	**occcc.ccc**	The details about the Collections Configuration Center application image. • **deadlineSeconds**: The maximum time, in seconds, for a deployment to make progress before it is considered failed. The default is **1200**. • **revisionHistLimit**: The maximum number of old ReplicaSets for this deployment to retain. The remaining is garbage-collected in the background. The default is **10**. • **imageName**: The name of the Collections Configuration Center image, such as **oracle/brm-collections-configuration-center**. • **imageTag**: The tag associated with the image. This is generally the release number prefixed with a colon (**:**). For example, **:16.26.0.0.0.0**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**. • **volMntKeyStore.***: The name and path of the volume mount containing the Collections Configuration Center KeyStore certificate. • **volMntAppExternalProperties.***: The name and path of the volume mount containing the application's external properties. • **volMntSecretEnv.***: The name and path of the volume mount that holds all passwords as a Secret. • **volMntLogs.***: The name and path of the volume mount where the log files are mounted.
probe.ready.*	**occcc.ccc.deployment**	The configuration for the readiness probe. • **delayInSec**: The duration, in seconds, to wait before performing the first readiness probe. The default is **200**. • **intervalInSec**: How often, in seconds, to perform the readiness probe. The default is **10**. • **maxAttempts**: The maximum number of consecutive failures before the probe is considered failed. The default is **100**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-3 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
configEnv.*	**occcc.ccc**	The configuration details for the Collections Configuration Center. • **name**: The name of this ConfigMap. The default is **brm-collections-configuration-center-env-configmap**. • **httpPort**: The container's port for accessing the application over HTTP. The default is **32000**. • **httpsPort**: The container's port for accessing the application over HTTPS. The default is **32001**. • **adminPort**: The administration port for health, metrics, and other administration-related activities. The default is **32080**. • **tlsVersions**: The list of TLS versions to support for connection with the WebLogic domain. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**. • **cccCertificateFileName**: The SSL certificate file name for Collections Configuration Center. • **trustStoreFileName**: This is the optional file name for the TrustStore. Set this key if the default Java TrustStore needs to be overridden. • **baseURL**: The base URL with resource details to return in the response of Collections Configuration Center requests. • **rsmURL**: The REST Service Manager connection properties where all Collections Configuration Center APIs are running. • **restartVersion**: The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**. • **securityEnabled**: The flag to indicate if token-based authentication is enabled for Collections Configuration Center (**true**) or not (**false**). The default is **true**. • **logLevel**: The logging level. The default is **INFO**. • **helidonSecurityLogLevel**: The Helidon security log level. The default is **INFO**. • **helidonWebServerLogLevel**: The Helidon Web Server log level. The default is **INFO**. • **helidonConfigLogLevel**: The Helidon configuration log level. The default is **INFO**. • **helidonCommonLogLevel**: The Helidon common log level. The default is **INFO**. • **auditLogLevel**: The audit log level. The default is **INFO**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-3 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
configEnv.oidc	**occcc.ccc**	The Identity Provider (IdP) authentication details. • **identity-uri**: The URI of the Identity Server, used as the base URL to retrieve metadata from the Identity Server. • **client-id**: The client ID generated by the Identity Server, used to validate the token. • **scope-audience**: The audience for the scope required by this application. This is prefixed to the scope name when requesting scopes from the Identity Server. • **audience**: The secondary audience configured in the IdP. If no secondary audience is configured, use the primary audience, which is the same as the scope-audience. • **header-use**: Whether the application extracts user identity information (such as user ID, email, or roles) from HTTP headers passed by IDCS. The default is **true**. • **frontend-uri**: The URI that is used to access the user-facing part of an application or website. • **server-type**: The IDCS integration mode or deployment environment used to validate and authenticate IDCS tokens. The default is **idcs**. • **logout-enabled**: Whether logging out of the application also logs the user out of IDCS. When set to **true**, the application redirects users to the IDCS logout endpoint, terminating their IDCS session. The default is **true**. • **post-logout-uri**: The URI that the client is redirected to post log out. If you define just a path, the host is taken from the header. The default is **/web/index.html**. • **logout-uri**: The URI that the client is redirected to at log out. If you define just a path, the host is taken from the header. The default is **/oauth2/v1/userlogout**. • **redirect**: Whether to redirect to the login page (and the token can be received either through a cookie or a header). The default is **true**. • **redirect-uri**: The endpoint URI where to redirect logins. The default is **/oidc/redirect**. • **cookie-http-only**: Whether the server reveals cookie information. The default is **true**. • **cookie-same-site**: The level of control when a browser sends cookies with cross-site requests to help prevent Cross-Site Request Forgery (CSRF) attacks: **STRICT**, **LAX**, or **NONE**. The default is **LAX**. • **access-token-ip-check**: Whether to verify an access token against an IP address for security purposes. The default is **false**. • **cookie-encryption-enabled**: Whether to encrypt cookies to protect them from being read or changed by third parties. The default is **false**.
secretKeyStore.extKeystoreSecret	**occcc.ccc**	The names of the pre-created Kubernetes Secrets for the Collections Configuration Center KeyStore certificates and wallets. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-3 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
secretVal.*	**occcc.ccc**	The credentials for accessing the system. • **name**: The name of the Kubernetes Secret that copies certificates to the container. The default is **brm-collections-configuration-center-env-secret**. • **cccCertificatePassword**: The Base64-encoded certificate password for Collections Configuration Center. • **trustStorePassword**: This is the optional file name for the TrustStore. Set this key if the default Java TrustStore needs to be overridden. • **clientSecret**: The Base64-encoded IDCS client secret.
hpa.*	**occcc.ccc**	The details for scaling up or down the number of pod replicas in your deployment based on a pod's CPU or memory utilization. By default, the Horizontal Pod Autoscaler is disabled. See "Setting Up Autoscaling of BRM Pods" in *BRM Cloud Native System Administrator's Guide*.
service.*	**occcc.ccc**	The brm-collections-configuration-center-service service's details. • **name**: The name of the service: **brm-collections-configuration-center-service**. • **type**: The service type. The default is **ClusterIP**.

Configuring Billing Care

Billing Care is a web-based client application that CSRs use to manage billing, payments, and accounts receivable for your customers. For more information about using Billing Care, see *Billing Care Online Help*.

To configure Billing Care to run in your BRM cloud native environment:

1. Override the Billing Care-specific keys from the **values.yaml** file for **oc-cn-op-job-helm-chart**. See "Adding Billing Care Keys for oc-cn-op-job-helm-chart".
2. Override the Billing Care-specific keys from the **values.yaml** file for **oc-cn-helm-chart**. See "Adding Billing Care Keys for oc-cn-helm-chart".
3. Set up volume mounts for Billing Care. See "About Billing Care Volume Mounts".
4. Create a WebLogic domain and install Billing Care. See "Creating a WebLogic Domain and Installing the Billing Care Application".
5. Set up SAML for SSO in Billing Care. See "Setting Up SSO for Billing Care".
6. Set up local users and groups for Billing Care. See "Setting Up Local Users and Groups for Billing Care".
7. Start and stop your WebLogic servers. See "Starting and Stopping WebLogic Servers".

> **Note**
>
> To set up Billing Care, ensure that you successfully complete the installation of **oc-cn-op-job-helm-chart** before you install or upgrade **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Adding Billing Care Keys for oc-cn-op-job-helm-chart

Table 9-4 lists a few important keys that directly impact Billing Care. Add these keys to your **override-values.yaml** file for **oc-cn-op-job-helm-chart** with the same path hierarchy.

For the complete set of keys to personalize your Billing Care deployment, see the keys with the path **ocbc.bc** in the **oc-cn-op-job-helm-chart/values.yaml** file.

> ⚠ **Caution**
>
> Keys with the path **ocbc.bc.secretVal** hold sensitive data. Handle them carefully with controlled access to the override file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Table 9-4 Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
isEnabled	**ocbc.bc**	Whether to deploy, configure, and start Billing Care services (**true**) or not (**false**). The default is **true**.
deployment.app.*	**ocbc.bc**	The details about the Billing Care application image. • **imageName**: The name of the Billing Care image, such as **oracle/billingcare**. • **imageTag**: The tag associated with the image. This is generally the release number prefixed with a colon (**:**). For example, **:16.26.0.0.0**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
deployment.fmw.*	**ocbc.bc**	The details about the Fusion Middleware Infrastructure image. • **imageRepository**: The name of the repository from where the Fusion Middleware Infrastructure image is pulled. The default is **container-registry.oracle.com/**. • **imageName**: The name of the Fusion Middleware Infrastructure image, such as **middleware/fmw-infrastructure_cpu**. • **imageTag**: The tag associated with the image. For example: **:14.1.2.0-jdk21-ol9**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.
deployment.sdk.*	**ocbc.bc**	The details about the Billing Care SDK image. • **imageName**: The name of the Billing Care SDK image. • **imageTag**: The tag associated with the image. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.
sdk.*	**ocbc.bc**	The details for deploying the Billing Care SDK. • **isEnabled**: Whether to deploy your SDK customizations for overriding application behavior (**true**) or not (**false**). The default is **false**. • **deployName**: The name of the Billing Care SDK to appear in the deployment list.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
configEnv.*	**ocbc.bc**	The details about the Managed Server. • **managedHttpPort**: The container's port for access to the Managed Server. The default is **8001**. • **httpPort**: The container's port for access to the WebLogic Domain over HTTP. The default is **7011**. • **serverStartMode**: The mode to use when starting the server: development mode (**dev**) or production mode (**prod**). The default is **prod**. • **adminUser**: The user who will be granted administrator rights to the WebLogic Domain. • **dbSSLMode**: The type of connection required to connect to the database: one-way SSL authentication (**ONE_WAY**) or SSL authentication is not required (**no**). • **dbWalletType**: The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**. • **extDBSSLWalletSecret**: The names of the pre-created Kubernetes Secrets for the Billing Care KeyStore certificate file. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
RCU Schema	**ocbc.bc.configEnv**	The details about the RCU schema. • **rcuJdbcURL**: The connection string for the database where schemas needed by Oracle Fusion Middleware products will be created, especially OPSS. • **rcuSysDBAUser**: The database administrator user name. • **rcuDBARole**: The role of the database administrator user. • **rcuPrefix**: The prefix for the OPSS schema. The default is **BC01**. • **rcuRecreate**: Whether to drop the existing OPSS schema (**true**) or not (**false**). The default is **true**. • **rcuArgs**: The additional arguments for the RCU schema. • **rcuTablespace**: The name of an existing tablespace in your database. If left empty, new tablespaces are created with names starting with **rcuPrefix**. • **rcuTempTablespace**: The name of an existing temporary tablespace in your database. If left empty, new tablespaces are created with names starting with **rcuPrefix**. • **isOPSS**: Whether to create an OPSS domain (**true**) or a non-OPSS domain (**false**). The default is **true**. • **extAccessPolicyCM**: The name of the ConfigMap containing the policy file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
LDAP Server	**ocbc.bc.configEnv**	The details about the LDAP Server. • **isLDAPEnabled**: Whether to skip creation of the Oracle Unified Directory Authenticator (**true**) or not (**false**). The default is **true**. • **ldapAdmin**: The Distinguished Name to connect to the LDAP server. • **ldapHost**: The host name or IP address of the LDAP Server (for example, OUD) where users and groups will be configured for access to Billing Care. • **ldapPort**: The port number on which the LDAP server is listening. • **ldapGroupBase**: The LDAP base DN that contains groups. • **ldapUserBase**: The LDAP base DN that contains users. • **ldapProviderName**: The name of Authentication Provider. • **targetServer**: The server in the WebLogic domain where the application must be deployed.
KeyStore Certificates	**ocbc.bc.configEnv**	The details about the KeyStore certificates for Billing Care. • **keystoreAlias**: The private key alias of the KeyStore. • **extKeystoreSecret**: The name of the pre-created external Secret containing the Identity and Trust KeyStore certificate file. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **keystoreType**: The file type of the SSL Identity and TrustStore certificate, which is either **PKCS12** or **JKS**. The default is **PKCS12**. • **keystoreIdentityFileName**: The file name of the Identity KeyStore certificate file. • **keystoreTrustFileName**: The file name of the TrustStore certificate file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
Secure Connections	**ocbc.bc.configEnv**	The details for secure connections. • **tlsVersions**: The list of TLS versions to support for connection with the WebLogic domain. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**. • **isSSOEnabled**: Whether to enable single sign-on (SSO) for Billing Care cloud native services through SAML 2.0 (**true**) or SSO is disabled (**false**). The default is **false**. • **extMetadataCM**: The name of the ConfigMap containing the IDP metadata file. • **samlAsserterName**: The name of the SAML Asserter. The default is **samlBCAsserter**. • **ssoPublishedSiteURL**: The base URL that is used to construct endpoint URLs. This is typically the Load Balancer host and port at which the server is visible externally. It must be appended with **/saml2**. For example: **https://***LoadBalancerHost***:***LoadBalancerPort***/saml2**. • **ssoDefaultURL**: The URL where unsolicited authentication responses are sent if they do not contain an accompanying target URL. • **reloadVersion**: Set this to any value different from the current value to force a restart of the deployer. The default is **1**. • **reset**: Whether to wipe all previous states and do a fresh setup of the domain (**true**) or not (**false**). The default is **false**. When set to **true**, you must change the **introspectVersion** key in the **oc-cn-helm-chart** must after upgrading **oc-cn-op-job-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
secretVal.*	**ocbc.bc**	The credentials for accessing the system. • **adminPassword**: The password of the WebLogic domain's administrative user, which is used for accessing the WebLogic Console for administrative operations. • **ldapPassword**: The password of the LDAP Server admin user. • **rcuSysDBAPassword**: The password for the rcuJdbcURL database administrator. • **rcuSchemaPassword**: The passwords for the schemas of Oracle Fusion Middleware products that will be created by RCU, which is used by OPSS. • **dbWalletPassword**: The password for accessing the certificates from the TrustStore and KeyStore. • **keystoreIdentityPassword**: The StorePass for the Identity KeyStore. • **keystoreKeyPassword**: The KeyPass for the Identity KeyStore. • **keystoreTrustPassword**: The StorePass for the Trust KeyStore.
secret.*	**ocbc.bc**	The details about the Billing Care external Kubernetes secrets: • **extFMWUserSecret**: The name of the external Kubernetes Secret containing the Fusion Middleware user passwords. • **extSetupSecret**: The name of the external Kubernetes Secret containing the setup passwords. • **extWLSUsersSecret**: The name of the external Kubernetes Secret containing the WebLogic Server user passwords.
monitoring.isEnabled	**ocbc.bc**	Whether monitoring is enabled in Billing Care. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
wop.*	**ocbc.bc**	The details about the WebLogic Domain. • **domainUID**: The name of the domain, which is used as a prefix to tag related objects. The default is **billingcare-domain**. • **domainRootDir**: The location within the container where the domain is created. The default is **/shared**. • **totalManagedServers**: The total number of managed servers forming the cluster. The default is **5**. • **initialServerCount**: The number of managed servers to initially start for the domain. The default is **2**. • **adminChannelPort**: The NodePort where the admin-server's HTTP service is accessible. The default is **30711**. • **serverStartPolicy**: The WebLogic servers that the Operator starts when it discovers the domain: – **NEVER**: Does not start any server in the domain. – **ADMIN_ONLY**: Starts only the administration server (no managed servers will be started). – **IF_NEEDED**: Starts the administration server and clustered servers up to the replica count. This is the default.
domain.*	**ocbc.bc.volume**	Details about the PVC for the domain file system. • **storage**: The storage size of the volume. • **createOption**: By default, the billingcare pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-4 (Cont.) Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
batchPayment.*	**ocbc.bc.volume**	Details about the PVC for the batch payment files. • **storage**: The storage size of the volume. • **createOption**: By default, the billingcare pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.
wlsUserGroups.*	**ocbc.bc**	The details for adding users and groups to the domain's default authenticator.
extensions.*	**ocbc.bc**	The name of the ConfigMap containing scripts to execute additional steps to configure the domain and application.
resources.*	**ocbc.bc**	The minimum and maximum CPU and memory resources that containers can use. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
nodeSelector	**ocbc.bc**	The node selector rules for scheduling WebLogic Server pods on particular nodes using simple selectors.
affinity	**ocbc.bc**	The affinity rules for scheduling WebLogic Server pods on particular nodes using more powerful selectors.
addOnPodSpec	**ocbc.bc**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Adding Billing Care Keys for oc-cn-helm-chart

Table 9-5 lists a few important keys that directly impact Billing Care. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart** with the same path hierarchy.

For the complete set of keys to personalize your Billing Care deployment, see the keys with the path **ocbc.bc** in the **oc-cn-helm-chart/values.yaml** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Keys with the path **ocbc.bc.secretVal** hold sensitive data. Handle them carefully with controlled access to the override file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Table 9-5 Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
appLogLevel	**ocbc**	The logging level at which application logs must be captured in log files: **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, **FINEST**, and **ALL**.
isEnabled	**ocbc.bc**	Whether to deploy, configure, and start Billing Care services (**true**) or not (**false**). The default is **true**.
deployment.app.*	**ocbc.bc**	The details about the Billing Care application image. • **imageName**: The name of the Billing Care image, such as **oracle/billingcare**. • **imageTag**: The tag associated with the image. This is generally the release number prefixed with a colon (**:**). For example, **:16.26.0.0.0**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.
deployment.fmw.*	**ocbc.bc**	The details about the Fusion Middleware Infrastructure image. • **imageRepository**: The name of the repository from where the Fusion Middleware Infrastructure image is pulled. The default is **container-registry.oracle.com/**. • **imageName**: The name of the Fusion Middleware Infrastructure image, such as **middleware/fmw-infrastructure**. • **imageTag**: The tag associated with the image. For example: **:14.1.2.0-jdk21-ol9**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-5 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
deployment.sdk.*	**ocbc.bc**	The details about the Billing Care SDK image. • **imageName**: The name of the Billing Care SDK image. • **imageTag**: The tag associated with the image. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**.
sdk.*	**ocbc.bc**	The details for deploying the Billing Care SDK. • **isEnabled**: Whether to deploy your SDK customizations for overriding application behavior (**true**) or not (**false**). The default is **false**. • **deployName**: The name of the Billing Care SDK to appear in the deployment list.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-5 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
configEnv.*	**ocbc.bc**	The details about the Managed Server. • **httpPort**: The container's port for access to the WebLogic Domain over HTTP. The default is **7011**. • **isOPSS**: Whether to create an OPSS domain (**true**) or a non-OPSS domain (**false**). The default is **true**. • **keystoreAlias**: The private key alias of the KeyStore. • **extKeystoreSecret**: The name of the pre-created external Secret containing the Identity and Trust KeyStore certificate file. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **keystoreType**: The file type of the SSL Identity and TrustStore certificate, which is either **PKCS12** or **JKS**. The default is **PKCS12**. • **dbSSLMode**: The type of connection required to connect to the database: one-way SSL authentication (**ONE_WAY**) or SSL authentication is not required (**no**). • **dbWalletType**: The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**. • **extDBSSLWalletSecret**: The names of the pre-created Kubernetes Secrets for the Billing Care KeyStore certificate file. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*. • **bipUrl**: The URL to the BI Publisher server. • **bipUserId**: The name of the user with access to the BI Publisher instance.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-5 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
infranet.*	**ocbc.bc**	The details for connecting to BRM. • **user.***: Information about the user having permissions to access BRM. • **connectionpool.***: The details about the connection pool. The default minimum is **25** and the default maximum is **50**. • **logLevel**: The log level for the **infranet.properties** file. • **ssoLogoutURL**: The URL where the user is redirected after logging out from the application. • **addOnProperties**: This key is empty by default. You can use this key to specify custom **Infranet.properties** values.
secretVal.*	**ocbc.bc**	The credentials for accessing the system. • **walletPassword**: The password of the wallet storing sensitive data for the BRM connection. • **bipPassword**: The password of the BI Publisher instance. • **keystoreTrustPassword**: The StorePass for the Trust KeyStore.
secret.*	**ocbc.bc**	The details about the Billing Care external Kubernetes secrets: • **extAppSecret**: The name of the external Kubernetes secret containing the Billing Care application passwords. • **extFMWUserSecret**: The name of the external Kubernetes secret containing the Fusion Middleware user passwords.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-5 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
wop.*	**ocbc.bc**	The details about the WebLogic Domain. • **domainUID**: The name of the domain, which is used as a prefix to tag related objects. The default is **billingcare-domain**. • **domainRootDir**: The location within the container where the domain is created. The default is **/shared**. • **totalManagedServers**: The total number of managed servers forming the cluster. The default is **5**. • **initialServerCount**: The number of managed servers to initially start for the domain. The default is **2**. • **adminChannelPort**: The NodePort where the admin-server's HTTP service is accessible. **Note:** Set this key only if you want the **billingcare-domain-admin-server-ext** service to deploy as NodePort. • **serverStartPolicy**: The WebLogic servers that the Operator starts when it discovers the domain: – **NEVER**: Does not start any server in the domain. – **ADMIN_ONLY**: Starts only the administration server (no managed servers will be started). – **IF_NEEDED**: Starts the administration server and clustered servers up to the replica count. This is the default. • **restartVersion**: Whether to force a rolling restart of all server pods. Change to any value other than current to trigger the action. • **introspectVersion**: Whether to force a domain introspection on change in domain configuration. Change to any value other than current to trigger the action.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-5 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
monitoring.*	**ocbc.bc**	Details about monitoring Billing Care. • **isEnabled**: Whether to enable monitoring of Billing Care (**true**) or not (**false**). See "Monitoring and Autoscaling Billing Care Cloud Native" in *BRM Cloud Native System Administrator's Guide*. • **imageRepository**: By default, the billingcare pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*. • **imageName**: The name of the WebLogic Monitoring Exporter image. The default is **oracle/ weblogic-monitoring-exporter**. • **imageTag**: The tag associated with the image. The default is **:2.2.2**. • **imagePullPolicy**: When to pull images: only when one is not present locally (**IfNotPresent**) or always (**Always**). The default is **IfNotPresent**. • **queries**: The WebLogic Monitoring Exporter query configuration that specifies the WebLogic Server runtime MBeans and attributes from which metrics are collected. By default, this key is empty. • **scrapeInterval**: The duration at which Prometheus scrapes the target. The default is **2s**. • **operator.isEnabled**: Whether the system is using Prometheus Operator and ServiceMonitor to scrape metrics (**true**) or not (**false**). The default is **false**. • **resources.***: The minimum and maximum CPU and memory resources that containers can use. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
extensions.scriptsConfigName	**ocbc.bc**	The name of the ConfigMap containing scripts to execute additional steps to configure the domain and application.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-5 (Cont.) Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
resources.*	**ocbc.bc**	The minimum and maximum CPU and memory resources that containers can use. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
nodeSelector	**ocbc.bc**	The node selector rules for scheduling WebLogic Server pods on particular nodes using simple selectors.
affinity	**ocbc.bc**	The affinity rules for scheduling WebLogic Server pods on particular nodes using more powerful selectors.
addOnPodSpec	**ocbc.bc**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Updating Infranet.properties for Billing Care

The **Infranet.properties** file entries are located in the **values.yaml** file. This makes it easier to update them.

Following is a sample configuration block (located in the **ocbc.bc** path in **oc-cn-helm-chart**) for the **Infranet.properties** entries:

```
infranet:
    user:
        login: 'boc_client.0.0.0.1'
        serviceType: '/service/admin_client'
        serviceId: 2
    connectionpool:
        minSize: 25
        maxSize: 50
    logLevel: 3
    ssoLogoutURL:
    addOnProperties: ""
```

If you have custom field classes, they should be provided through the SDK **.war** file and defined here using the **addOnProperties** key. For example:

```
addOnProperties:|-
    infranet.custom.field.package=com.portal.custom
    infranet.custom.field.100011=PIN_FLD_ABC
```

To update these properties, update the values in **override-values.yaml** file for **oc-cn-helm-chart**. If this is an upgrade, also update the **ocbc.bc.wop.restartVersion** key in the same file. This will force a pod restart and the new values will be used.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Adding Custom Configuration to Deployment Workflow for Billing Care

You can provide additional configuration to be applied at particular checkpoints in the Billing Care deployment workflow. These checkpoints are:

- **ext_deployer_pre_exit**: Called after the standard configuration in **deployer.sh** in **oc-cn-op-job-helm-chart**
- **ext_init_app_pre_exit**: Called after the standard configuration in the init-app **initContainer** container in both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**
- **ext_init_config_pre_exit**: Called after the standard configuration in the init-config **initContainer** container in both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**

Create a ConfigMap with your configuration scripts, including a shell script named **run_hooks.sh** that calls your other scripts. For example:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ext-scripts
data:
  run_hooks.sh: |+
    #!/bin/bash
    echo "executing extension for: $@"
    CURRENT_CHECKPOINT=$1
    if [ "$CURRENT_CHECKPOINT" == "ext_deployer_pre_exit" ] ; then
      sh my_deployer_extension.sh
    fi
  my_deployer_extension.sh: |+
    #!/bin/bash
    echo "executing my_deployer_extension"
...
```

Specify the name of your ConfigMap in the **ocbc.bc.extensions.scriptsConfigName** key in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

Since Billing Care is a web application that is deployed on WebLogic Server, refer to the WebLogic Server documentation for information about overriding timeouts, cookie attributes, and so on. See "web.xml Deployment Descriptor Elements" and "weblogic.xml Deployment Descriptor Elements" in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server* for more information about these configurations. You can find files to help you with this configuration in the **oc-cn-op-job-helm-chart/templates** directory.

About Billing Care Volume Mounts

The Billing Care container requires Kubernetes volume mounts for sharing the domain and application file system between the WebLogic Cluster servers. There is one volume for the domain and one for batch payments. By default, these are created dynamically, using the provisioner defined in BRM, in the **storage-class** key in **oc-cn-op-job-helm-chart**.

To change the volume type or provider, modify the following keys in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

- **ocbc.bc.volume.domain.createOption** for the domain file system for Billing Care.
- **ocbc.bc.volume.batchPayment.createOption** for the batch payments file system.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Creating a WebLogic Domain and Installing the Billing Care Application

The WebLogic domain is created by a Kubernetes Deployment when **oc-cn-op-job-helm-chart** is installed. The same job also installs the Billing Care application and deploys the application WAR file onto the WebLogic Cluster.

The **oc-cn-op-job-helm-chart** chart also:

- Creates a Kubernetes ConfigMap and Secrets, which are used throughout the life-cycle of the WebLogic domain.
- Initializes the **PersistentVolumeClaim** for the domain and application file system as well as third-party libraries.

> **Note**
>
> The **override-values.yaml** file that you use for this chart must include BRM override values.

After you install **oc-cn-op-job-helm-chart**, wait until the Kubernetes deployment has reached the **1/1 Running** status. Then, you can install or upgrade **oc-cn-helm-chart** for Billing Care services.

After the deployment is running, don't delete the chart. Its resources will be used for starting and stopping the servers through **oc-cn-helm-chart**.

Setting Up SSO for Billing Care

SSO allows users to log in to applications using a single user name and password combination. You set up SSO for Billing Care cloud native services by using SAML 2.0.

To set up SSO for Billing Care:

1. Export the SAML 2.0 metadata XML file from your identity and access management (IAM) system.

   For example, if you are using Oracle Access Management, you can export the file by following the instructions in "Exporting Metadata" in *Oracle Fusion Middleware Administering Oracle Access Management*.

2. Add the metadata XML file to your BRM cloud native deployment by doing one of the following:
   - Rename the metadata XML file to **metadata.xml**, and then move **metadata.xml** to the **oc-cn-op-job-helm-chart/billingcare/idp** directory.
   - Pre-create the IDP metadata ConfigMap for Billing Care and set the **ocbc.bc.configEnv.extMetadataCM** key in your **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

     For more information, see "Managing Wallet and KeyStore Certificates" in *BRM Cloud Native System Administrator's Guide*.

3. Configure the KeyStores needed by SAML by doing one of the following:
   - Generate the Identity and Trust KeyStores and then move your files, such as **identity.p12** and **trust.p12**, under the **oc-cn-op-job-helm-chart/billingcare/keystore** directory.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Pre-create the Kubernetes Secret for the Identity and Trust KeyStore files and set the **ocbc.bc.configEnv.extKeystoreSecret** key in your **override-values.yaml** file for both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**.

  For more information, see "Managing Wallet and KeyStore Certificates" in *BRM Cloud Native System Administrator's Guide*.

4. In your **override-values.yaml** file for **oc-cn-op-job-helm-chart**, set the following keys:
   - **ocbc.bc.configEnv.isSSOEnabled**: Set this to **true**.
   - **ocbc.bc.configEnv.keystoreAlias**: Set this to the private key alias of the KeyStore.
   - **ocbc.bc.configEnv.keystoreType**: Set this to the file type of the SSL Identity and Trust store, which is either **PKCS12** or **JKS**. The default is **PKCS12**.
   - **ocbc.bc.configEnv.keystoreIdentityFileName**: Set this to the name of the Identity KeyStore file.
   - **ocbc.bc.configEnv.keystoreTrustFileName**: Set this to the name of the Trust KeyStore file.
   - **ocbc.bc.configEnv.samlAsserterName**: Set this to the name of the SAML Asserter. The default is **samlBCAsserter**.
   - **ocbc.bc.configEnv.ssoPublishedSiteURL**: Set this to the base URL that is used to construct endpoint URLs. This is typically the load balancer host and port at which the server is visible externally. It must be appended with **/saml2**. For example: **https://** *LoadBalancerHost*:*LoadBalancerPort*/**saml2**.
   - **ocbc.bc.configEnv.ssoDefaultURL**: Set this to the URL where unsolicited authentication responses are sent if they do not contain an accompanying target URL.
   - **ocbc.bc.secretVal.keystoreIdentityPassword**: Set this to the StorePass for the Identity KeyStore.
   - **ocbc.bc.secretVal.keystoreKeyPassword**: Set this to the KeyPass for the Identity KeyStore.
   - **ocbc.bc.secretVal.keystoreTrustPassword**: Set this to the StorePass for the Trust KeyStore.
5. Configure your load balancer's rules to send responses to the Billing Care WebLogic domain with **/saml2** appended to the URL path.

   > **Note**
   >
   > Add this rule to your existing load balancer rules for routing responses to Billing Care (**/bc**), the load balancer host name, and so on.

   See "Exposing BRM Services Outside the Kubernetes Cluster".
6. Deploy your Billing Care cloud native services by following the instructions in "Deploying BRM Cloud Native Services".
7. After Billing Care is deployed, retrieve the **sp-metadata-admin-server.xml** file from the **/shared/domains/***domainUID* directory in your container, where *domainUID* is the name of your Billing Care domain specified in the **ocbc.bc.wop.domainUID** key.

   The XML file configures the Web SSO Provider Partner. It contains the partner's KeyStore certificates, SAML assertion details, and the URLs where the SAML Identity Provider redirects to provide access to Billing Care.

Copyright © 2019, 2026, Oracle and/or its affiliates.

8. Create a profile for your identity provider partner by loading the **sp-metadata-admin-server.xml** file into your IAM system.

   For example, if you are using Oracle Access Management, you can load the file by following the instructions in "Creating Remote Identity Provider Partners" in *Oracle Fusion Middleware Administering Oracle Access Management*.

Setting Up Local Users and Groups for Billing Care

You have the option to customize the values for **oc-cn-op-job-helm-chart** to create users and groups locally in Oracle WebLogic Server. This would be especially useful for test environments where you might not have Identity Providers or LDAPs available. The groups for the admin user for WebLogic Server cannot be modified using this procedure.

Any passwords must be encoded using Base64. You can leave the password blank, but then the user will not be able to log in to the application directly.

To set up local users and groups for Billing Care, define the keys under **ocbc.bc.wlsUserGroups** in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**. For example:

```
ocbc:
    bc:
        wlsUserGroups:
            groups:
            -   name: "GroupA"
                description: "GroupA Description"
            -   name: "GroupB"
                description: "GroupB Description"
            users:
            -   name: csr1
                description: "csr1 description"
                password: "Base64_password"
                groups:
                -   "GroupA"
                -   "GroupB"
            -   name: csr2
                description: "csr2 description"
                password: "Base64_password"
                groups:
                -   "GroupB"
```

Starting and Stopping WebLogic Servers

When you install **oc-cn-op-job-helm-chart**, the default configuration sets up a WebLogic Cluster with five Managed Servers. When you install or upgrade **oc-cn-helm-chart** for the Billing Care service, two of the Managed Servers and one Admin Server are started.

By modifying the **override-values.yaml** file for **oc-cn-helm-chart**, you can control:

- The total number of Managed Servers and the initial server start up by using the **totalManagedServers** and **initialServerCount** keys.
- Whether the servers are started or stopped by using the **serverStartPolicy** key. To start the Admin Servers and the Managed Servers in a Cluster, set the key to **IF_NEEDED**. To stop all servers, set the key to **NEVER**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> The keys in the **override-values.yaml** file should be the same as the ones used in **oc-cn-op-job-helm-chart** for keys that are common in both charts.

After you modify the **override-values.yaml** file, update the Helm release for the changes to take effect:

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
```

where:

- *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.
- *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.
- *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-helm-chart**.

Configuring the Billing Care REST API

You use the Billing Care REST API to integrate an external customer management application with BRM. This allows you to manage billing and rating in BRM and then manage your customers' accounts and bills in your external application. For more information, see *REST API Reference for Billing Care*.

To configure the Billing Care REST API to work with BRM cloud native:

1. Override the Billing Care REST API-specific keys from the **values.yaml** file for **oc-cn-op-job-helm-chart**. See "Adding Billing Care REST API Keys for oc-cn-op-job-helm-chart".
2. Override the Billing Care REST API-specific keys from the **values.yaml** file for **oc-cn-helm-chart**. See "Adding Billing Care REST API Keys for oc-cn-helm-chart".
3. Set up volume mounts for the Billing Care REST API. See "About Billing Care REST API Volume Mounts".
4. Create a WebLogic domain and install the Billing Care REST API. See "Creating a WebLogic Domain and Installing the Billing Care REST API".
5. Set up local users and groups for Billing Care REST API. See "Setting Up Local Users and Groups for Billing Care REST API".
6. Start and stop your WebLogic servers. See "Starting and Stopping WebLogic Servers".

> **Note**
>
> To set up the Billing Care REST API, ensure that you successfully complete the installation of **oc-cn-op-job-helm-chart** before you install or upgrade **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Adding Billing Care REST API Keys for oc-cn-op-job-helm-chart

Table 9-6 lists a few important keys that directly impact the Billing Care REST API. Add these keys to your **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

For the complete set of keys to personalize your Billing Care REST API deployment, see the keys with the path **ocbc.bcws** in the **oc-cn-op-job-helm-chart/values.yaml** file.

> ⚠ **Caution**
>
> Keys with the path **ocbc.bcws.secretVal** hold sensitive data. Handle them carefully with controlled access to the override file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Table 9-6 Billing Care REST API Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
isEnabled	**ocbc.bcws**	Whether to deploy, configure, and start Billing Care REST API services: • **false**: Does not create the Kubernetes resources for using the Billing Care REST API. • **true**: Creates the Kubernetes resources for using the Billing Care REST API. This is the default.
imageName	**ocbc.bcws.deployment.app**	The name of the Billing Care REST API image, such as **oracle/bcws**.
imageTag	**ocbc.bcws.deployment.app**	The tag associated with the image. This is generally the release number. Prefix the value with a colon (:). For example, **:16.26.0.0.0.0**.
dbSSLMode	**ocbc.bcws.configEnv**	The type of connection required to connect to the database: • **TWO_WAY**: Two-way SSL authentication is required. In this case, both the client and server must authenticate each other's identity. • **ONE_WAY**: One-way SSL authentication is required. In this case, the client must authenticate the server's identity. This is the default. • **NO**: SSL authentication is not required.
dbWalletType	**ocbc.bcws.configEnv**	The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**.
extDBSSLWalletSecret **extKeystoreSecret**	**ocbc.bcws.configEnv**	The names of the pre-created Kubernetes Secrets for the Billing Care REST API KeyStore certificates and wallets. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
dbWalletPassword	**ocbc.bcws.configEnv**	The password for accessing the certificates from the TrustStore and KeyStore.
rcuJdbcURL	**ocbc.bcws.configEnv**	The connection string for connecting to the database where schemas needed by Oracle Fusion Middleware products will be created, especially OPSS.
rcuDBARole	**ocbc.bcws.configEnv**	The role of the database administrator user.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-6 (Cont.) Billing Care REST API Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
rcuArgs	**ocbc.bcws.configEnv**	The additional arguments for creating the RCU.
ldapHost	**ocbc.bcws.configEnv**	The host name or IP address of the LDAP Server (for example, OUD) where users and groups will be configured for access to the Billing Care REST API.
ldapPort	**ocbc.bcws.configEnv**	The port number on which the LDAP server is listening.
ldapGroupBase	**ocbc.bcws.configEnv**	The LDAP base DN that contains groups.
ldapUserBase	**ocbc.bcws.configEnv**	The LDAP base DN that contains users.
keystoreAlias	**ocbc.bcws.configEnv**	The private key alias of the KeyStore.
keystoreType	**ocbc.bcws.configEnv**	The file type of SSL Identity and Trust store, either **PKCS12** or **JKS**.
keystoreIdentityFileName	**ocbc.bcws.configEnv**	The file name of the Identity KeyStore.
keystoreTrustFileName	**ocbc.bcws.configEnv**	The file name of the Trust KeyStore.
tlsVersions	**ocbc.bcws.configEnv**	The list of TLS versions to support for connection with the WebLogic domain. List the version numbers in order, from lowest to highest, separated by a comma. For example: **TLSv1.2, TLSv1.3**.
reloadVersion	**ocbc.bcws.configEnv**	Update this value with any value different from the current value to force a restart of the deployer.
adminPassword	**ocbc.bcws.secretVal**	The password of the WebLogic domain's administrative user, which is used for accessing the WebLogic Console for administrative operations.
ldapPassword	**ocbc.bcws.secretVal**	The password of the LDAP Server admin user.
rcuSysDBAPassword	**ocbc.bcws.secretVal**	The password for the rcuJdbcURL database administrator.
rcuSchemaPassword	**ocbc.bcws.secretVal**	The passwords for the schemas of Oracle Fusion Middleware products that will be created by RCU, which is used by OPSS.
dbWalletPassword	**ocbc.bcws.secretVal**	The password for accessing the certificates from the TrustStore and KeyStore.
keystoreIdentityPassword	**ocbc.bcws.secretVal**	The storepass of the Identity KeyStore.
keystoreKeyPassword	**ocbc.bcws.secretVal**	The KeyPass of the Identity KeyStore.
keystoreTrustPassword	**ocbc.bcws.secretVal**	The storepass of Trust KeyStore.
domainUID	**ocbc.bcws.wop**	The name of the domain. The default is **bcws-domain**.
adminChannelPort	**ocbc.bcws.wop**	The NodePort where the admin-server's HTTP service is accessible. The default is **30711**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-6 (Cont.) Billing Care REST API Keys for oc-cn-op-job-helm-chart

Key	Path in values.yaml File	Description
serverStartPolicy	**ocbc.bcws.wop**	The WebLogic servers that the Operator starts when it discovers the domain: • **NEVER**: Does not start any server in the domain. • **ADMIN_ONLY**: Starts only the administration server (no managed servers will be started). • **IF_NEEDED**: Starts the administration server and clustered servers up to the replica count.
domain.*	**ocbc.bcws.volume**	Details about the PVC for the domain file system: • **storage**: The storage size of the volume. • **createOption**: By default, the bcws pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.
batchPayment.*	**ocbc.bcws.volume**	Details about the PVC for the batch payment files: • **storage**: The storage size of the volume. • **createOption**: By default, the bcws pod uses dynamic volume provisioning. To use a static volume instead, you must add the **createOption** key. See "Using Static Volumes" in *BRM Cloud Native System Administrator's Guide*.
resources.*	**ocbc.bcws**	The minimum and maximum CPU and memory resources for the cm pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
nodeSelector	**ocbc.bcws**	The node selector rules for scheduling WebLogic Server pods on particular nodes using simple selectors.
affinity	**ocbc.bcws**	The affinity rules for scheduling WebLogic Server pods on particular nodes using more powerful selectors.
addOnPodSpec	**ocbc.bcws**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Adding Billing Care REST API Keys for oc-cn-helm-chart

Table 9-7 lists a few important keys that directly impact the Billing Care REST API. Add these keys to your **override-values.yaml** file for **oc-cn-helm-chart**.

For the complete set of keys to personalize your Billing Care REST API deployment, see the keys with the path **ocbc.bcws** in the **oc-cn-helm-chart/values.yaml** file.

> ⚠ **Caution**
>
> Keys with the path **ocbc.bcws.secretVal** hold sensitive data. Handle them carefully with controlled access to the override file containing their values. Encode all of these values in Base64 format. See "Secrets" in *Kubernetes Concepts*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-7 Billing Care REST API Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
appLogLevel	**ocbc**	The logging level at which application logs must be captured in log files: **SEVERE**, **WARNING**, **INFO**, **CONFIG**, **FINE**, **FINER**, **FINEST**, and **ALL**.
isEnabled	**ocbc.bcws**	Whether to deploy, configure, and start Billing Care REST API services: • **false**: Does not create the Kubernetes resources for using the Billing Care REST API. • **true**: Creates the Kubernetes resources for using the Billing Care REST API. This is the default.
imageName	**ocbc.bcws.deployment.app**	The name of the Billing Care REST API image, such as **oracle/bcws**.
imageTag	**ocbc.bcws.deployment.app**	The tag associated with the image. This is generally the release number. Prefix the value with a colon (:). For example, **:16.26.0.0.0.0**.
keystoreAlias	**ocbc.bcws.configEnv**	The private key alias of the KeyStore.
extDBSSLWalletSecret **extKeystoreSecret**	**ocbc.bcws.configEnv**	The names of the pre-created Kubernetes Secrets for the Billing Care REST API KeyStore certificates and wallets. See "Using External Kubernetes Secrets for Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.
dbSSLMode	**ocbc.bcws.configEnv**	The type of connection required to connect to the database: • **TWO_WAY**: Two-way SSL authentication is required. In this case, both the client and server must authenticate each other's identity. • **ONE_WAY**: One-way SSL authentication is required. In this case, the client must authenticate the server's identity. This is the default. • **NO**: SSL authentication is not required.
dbWalletType	**ocbc.bcws.configEnv**	The type of TrustStore and KeyStore file that is used for the SSL connection: **SSO** or **PKCS12**.
user.*	**ocbc.bcws.infranet**	The permissions for accessing BRM: • **login**: The user name of the service that has permission to access BRM. The default is **bc_client.0.0.0.1**. • **serviceType**: The POID type of the service that has permission to access BRM. The default is **/service/admin_client**. • **serviceID**: The POID ID of the service that has permission to access BRM. The default is **416**.
connectionpool.*	**ocbc.bcws.infranet**	The size of the connection pool • **minSize**: The minimum number of connections in the pool. The default is **25**. • **maxSize**: The maximum number of connections in the pool. The default is **50**.
loglevel	**ocbc.bcws.infranet**	The log level for the **Infranet.properties** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-7 (Cont.) Billing Care REST API Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
idp.*	**ocbc.bcws.infranet**	The details for the Identity Provider (IdP) for authenticating clients to access Billing Care REST API: • **vendor**: The IdP application: **OAM** or **IDCS**. • **url**: The base URL of the IdP server, such as https://host:port. • **resourceServerAndScope**: The resource server and scope created in your IDP for Billing Care REST API. • **identityDomain**: The name of the Identity Provider. This is mandatory if the vendor is OAM. • **clientId**: The client ID for accessing the IdP when validating the access token. This is mandatory if the vendor is IDCS.
addOnProperties	**ocbc.bcws.infranet**	This key is empty by default. You can use this key to specify custom **Infranet.properties** values.
secretVal.*	**ocbc.bcws**	The passwords: • **walletPassword**: The password for the wallet storing sensitive data for connecting to BRM. • **bipPassword**: The password for your BI Publisher instance. • **keystoreTrustPassword**: The StorePass for the Trust KeyStore. • **infranet.idp.clientSecret**: The client secret for communicating with the authorization server.
domainUID	**ocbc.bcws.wop**	The name of the domain. The default is **bcws-domain**.
adminChannelPort	**ocbc.bcws.wop**	The NodePort where the admin-server's HTTP service is accessible. By default, this key is blank. **Note**: Set this key only if you want the **bcws-domain-admin-server-ext** service to deploy as NodePort.
serverStartPolicy	**ocbc.bcws.wop**	The WebLogic servers that the Operator starts when it discovers the domain: • **NEVER**: Does not start any server in the domain. • **ADMIN_ONLY**: Starts only the administration server (no managed servers will be started). • **IF_NEEDED**: Starts the administration server and clustered servers up to the replica count. This is the default.
isEnabled	**ocbc.bcws.monitoring**	Whether to enable monitoring of Billing Care REST API. See "Monitoring and Autoscaling Billing Care Cloud Native" in *BRM Cloud Native System Administrator's Guide*.
resources.*	**ocbc.bcws**	The minimum and maximum CPU and memory resources for the cm pod. See "Setting Minimum and Maximum CPU and Memory Values" in *BRM Cloud Native System Administrator's Guide*.
nodeSelector	**ocbc.bcws**	The node selector rules for scheduling WebLogic Server pods on particular nodes using simple selectors.
affinity	**ocbc.bcws**	The affinity rules for scheduling WebLogic Server pods on particular nodes using more powerful selectors.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 9-7 (Cont.) Billing Care REST API Keys for oc-cn-helm-chart

Key	Path in values.yaml File	Description
addOnPodSpec	**ocbc.bcws**	The details for extending pod specifications or overriding features. By default, this key is empty. See "About Customizing and Extending Pods" in *BRM Cloud Native System Administrator's Guide*.

Updating Infranet Properties for the Billing Care REST API

The **Infranet.properties** file entries are located in the **values.yaml** file. This makes it easier to update them.

Following is a sample configuration block (located in the **ocbc.bcws** path in **oc-cn-helm-chart**) for the **Infranet.properties** entries:

```
infranet:
    user:
        login: 'bc_client.0.0.0.1'
        serviceType: '/service/admin_client'
        serviceId: 416
    connectionpool:
        minSize: 25
        maxSize: 50
    logLevel: 3
    idp:
        vendor: IDCS
        url: https://host:port
        resourceServerAndScope:
        identityDomain:
        clientId:
    addOnProperties: ""
```

If you have custom field classes, they should be provided through the SDK **.war** file and defined here using the **addOnProperties** key. For example:

```
addOnProperties:|-
    infranet.custom.field.package=com.portal.custom
    infranet.custom.field.100011=PIN_FLD_ABC
```

To update any of these properties after an install or upgrade, update the values in **override-values.yaml** file for **oc-cn-helm-chart**. If this is an upgrade, also update the **ocbc.bcws.wop.restartVersion** key in the same file. This will force a pod restart and the new values will be used.

Adding Custom Configuration to Deployment Workflow for Billing Care REST API

You can provide additional configuration to be applied at particular checkpoints in the Billing Care REST API deployment workflow. These checkpoints are:

- **ext_deployer_pre_exit**: Called after the standard configuration in **deployer.sh** in **oc-cn-op-job-helm-chart**

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **ext_init_app_pre_exit**: Called after the standard configuration in the init-app **initContainer** container in both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**
- **ext_init_config_pre_exit**: Called after the standard configuration in the init-config **initContainer** container in both **oc-cn-op-job-helm-chart** and **oc-cn-helm-chart**

Create a ConfigMap with your configuration scripts, including a shell script named **run_hooks.sh** that calls your other scripts. For example:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: ext-scripts
data:
  run_hooks.sh: |+
    #!/bin/bash
    echo "executing extension for: $@"
    CURRENT_CHECKPOINT=$1
    if [ "$CURRENT_CHECKPOINT" == "ext_deployer_pre_exit" ] ; then
      sh my_deployer_extension.sh
    fi
  my_deployer_extension.sh: |+
    #!/bin/bash
    echo "executing my_deployer_extension"
...
```

Specify the name of your ConfigMap in the **ocbc.bcws.extensions.scriptsConfigName** key in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

About Billing Care REST API Volume Mounts

The Billing Care REST API container requires Kubernetes volume mounts for sharing the domain and application file system between the WebLogic Cluster servers. There is one volume for the domain and one for batch payments. By default, these are created dynamically, using the provisioner defined in BRM, in the **storage-class** key in **oc-cn-op-job-helm-chart**.

> **Note**
>
> The selected location must be accessible on all worker nodes across which WebLogic Servers will be distributed based on defined nodeSelector or affinity rules.

To change the volume type or provider, modify the following keys in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

- **ocbc.bcws.volume.domain.createOption** for the domain file system for Billing Care.
- **ocbc.bcws.volume.batchPayment.createOption** for the batch payments file system.

Creating a WebLogic Domain and Installing the Billing Care REST API

The WebLogic domain is created by a Kubernetes Deployment when **oc-cn-op-job-helm-chart** is installed. The same job also installs the Billing Care REST API and deploys the application WAR file onto the WebLogic Cluster.

The **oc-cn-op-job-helm-chart** chart also:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Creates a Kubernetes ConfigMap and Secrets, which are used throughout the life-cycle of the WebLogic domain.
- Initializes the **PersistentVolumeClaim** for the domain and application file system as well as third-party libraries.

> **Note**
>
> The **override-values.yaml** file that you use for this chart must include BRM override values.

After you install **oc-cn-op-job-helm-chart**, wait until the Kubernetes deployment has reached the **1/1 Running** status. Then, you can install or upgrade **oc-cn-helm-chart** for Billing Care REST API services.

After the deployment is running, don't delete the chart. Its resources will be used for starting and stopping the servers through **oc-cn-helm-chart**.

Setting Up Local Users and Groups for Billing Care REST API

You have the option to customize the values for **oc-cn-op-job-helm-chart** to create users and groups locally in Oracle WebLogic Server. This would be especially useful for test environments where you might not have Identity Providers or LDAPs available. The groups for the admin user for WebLogic Server cannot be modified using this procedure.

Any passwords must be encoded using Base64. You can leave the password blank, but then the user will not be able to log in to the application directly.

To set up local users and groups for Billing Care, define the keys under **ocbc.bcws.wlsUserGroups** in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**. For example:

```
ocbc:
    bcws:
        wlsUserGroups:
            groups:
            -   name: "GroupA"
                description: "GroupA Description"
            -   name: "GroupB"
                description: "GroupB Description"
            users:
            -   name: csr1
                description: "csr1 description"
                password: "Base64_password"
                groups:
                -   "GroupA"
                -   "GroupB"
            -   name: csr2
                description: "csr2 description"
                password: "Base64_password"
                groups:
                -   "GroupB"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Starting and Stopping WebLogic Servers

When you install **oc-cn-op-job-helm-chart**, the default configuration sets up a WebLogic Cluster with five Managed Servers. When you install or upgrade **oc-cn-helm-chart** for the Billing Care REST API service, two of the Managed Servers and one Admin Server are started.

By modifying the **override-values.yaml** file for **oc-cn-helm-chart**, you can control:

- The total number of Managed Servers and the initial server start up by using the **totalManagedServers** and **initialServerCount** keys.
- Whether the servers are started or stopped by using the **serverStartPolicy** key. To start the Admin Servers and the Managed Servers in a Cluster, set the key to **IF_NEEDED**. To stop all servers, set the key to **NEVER**.

> **Note**
>
> The keys in the **override-values.yaml** file should be the same as the ones used in **oc-cn-op-job-helm-chart** for keys that are common in both charts.

After you modify the **override-values.yaml** file, update the Helm release for the changes to take effect:

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
```

where:

- *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.
- *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.
- *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

10

Configuring ECE Services

Learn how to configure Oracle Communications Elastic Charging Engine (ECE) services by configuring and deploying the ECE Helm chart.

Topics in this document:

- Adding Elastic Charging Engine Keys
- Enabling SSL in Elastic Charging Engine
- Connecting ECE Cloud Native to an SSL-Enabled Database
- About Elastic Charging Engine Volume Mounts
- Loading Custom Diameter AVP
- Generating CDRs for Unrated Events
- Configuring ECE to Support Prepaid Usage Overage
- Recording Failed ECE Usage Requests
- Loading BRM Configuration XML Files
- Setting Up Notification Handling in ECE
- Configuring ECE for a Multischema BRM Environment
- Configuring ECE for Overload Protection
- Configuring Direct Loading Using the ECE RE Manager Plug-in

For information about performing administrative tasks on your ECE cloud native services, see "Administering ECE Cloud Native Services" in *BRM Cloud Native System Administrator's Guide*.

Before installing the ECE Helm chart, you must first publish the metadata, config, and pricing data from the PDC pod.

> **Note**
>
> Kubernetes looks for the CPU limit setting for pods. If it's not set, Kubernetes allocates a default value of 1 CPU per pod, which causes CPU overhead and Coherence scalability issues. To prevent this from happening, override each ECE pod's CPU limit to be the maximum CPU available on the node.

Adding Elastic Charging Engine Keys

Table 10-1 lists the keys that directly impact ECE deployment. Add these keys to your **override-values.yaml** file for **oc-cn-ece-helm-chart**. In the table, *component-name* should be replaced with the name of the ECE component, such as **emgateway**, **radiusgateway**, **diametergateway**, **httpgateway**, and **ratedeventformatter**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
imageRepository	N/A	The repository server from where images must be pulled.
imagePullSecrets	N/A	The secrets required to pull images from the repository.
container.*	N/A	The details for the ECE container: • **image**: The image to be used in the container. The default is **oc-cn-ece:16.26.0.0.0**. • **imagePullPolicy**: The image pull policy setting specifies the configured value. The default value is **IfNotPresent**, which specifies to not pull the image if it's already present. Applicable values are **IfNotPresent** and **Always**.
TZ	N/A	The TZ setting specifies the configured value. The default is **UTC**.
chargingSettingPath	**volume**	The location of the configuration folder for ECE. The default is **/home/charging/opt/ECE/oceceserver/config**.
createWalletsAsSecrets	N/A	Whether to create keystore certificates and wallets as secrets during the ECE deployment process (**true**) or if they have been pre-created as Kubernetes secrets (**false**). See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
secretEnv.*	N/A	The secret environment settings are: • **extECESecret**: The name of the external Kubernetes secret for ECE. • **name**: The user-defined name for the secret. The default is **secret-env**. • **uniPass**: The universal password used to configure the ECE passwords. • **walletPassword**: The password for opening the ECE wallet. • **JMSQUEUEPASSWORD**: The password for the JMS queue, which is stored as jms.queue.notif.pwd in the wallet. • **RADIUSSHAREDSECRET**: The RADIUS shared secret, which is stored as radius.secret.pwd in the wallet. • **NOTIFYEVENTKEYPASS**: The key passphrase for notification events. • **BRMGATEWAYPASSWORD**: The BRM Gateway password used for CM access. • **PDCPASSWORD**: The WebLogic administration password used for PDC and JMS queue access. • **PDCKEYSTOREPASSWORD**: The password for the PDC keystore. • **ECEJMXSSLKEYSTOREPASSWORD**: The password for the private key and the JMX PKCS12 keystore. • **PERSISTENCEDATABASEPASSWORD.***: The schema number and password for the ECE persistence database. • **PERSISTENCEDBAPASSWORD.***: The schema number and password for the ECE persistence database administrator. • **PERSISTENCEDATABASEKEYPASS.***: The schema number and password for the ECE persistence database passkey. • **ECEHTTPGATEWAYSERVERSSLKEYSTOREPASSWORD**: The password for the HTTP Gateway server SSL keystore. • **ECEHTTPGATEWAYIDENTITYKEYSTOREPASSWORD**: The password for the HTTP Gateway identity keystore. • **ECEHTTPGATEWAYTRUSTSTOREPASSWORD**: The password for the HTTP Gateway truststore. • **ECEKAFKATRUSTSTOREPASSWORD**: The password for the truststore that ECE uses to trust Kafka server certificates. • **ECEKAFKAOAUTHCLIENTSECRET**: The OAuth 2.0 client secret used when Kafka OAuth authentication is enabled.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **ECEOAUTH2SYMMETRICKEY**: The symmetric key used to validate OAuth 2.0 tokens when oauth2Enabled is true. • **BRMDATABASEPASSWORD.***: The schema number and password for the BRM database. • **BRMDATABASEKEYPASS.***: The schema number and password for the BRM database passkey. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information.
jmx.*	N/A	The details for the JMX keystore: • **serverSslKeyStore**: The JMX PKCS12 keystore filename or absolute path inside the container. The default is **jmxServer.p12**. • **serverSslKeyStoreType**: The JMX keystore type. Only PKCS12 is supported. The default is **PKCS12**. • **extSecuritySecret**: The existing secret containing the keystore, its password, and both JMX credential files. When empty, the chart creates and manages this secret. • **boundarySystemAdmin**: The Base64-encoded boundarySystemAdmin password.
sslconnectioncertificates.*	N/A	The details for the SSL certificates: • **SSLENABLED**: Whether SSL is enabled in ECE (**true**) or not (**false**). • **DNAME**: The domain name. For example: "CN=Admin, OU=Oracle Communication Application, O=Oracle Corporation, L=Redwood Shores, S=California, C=US" • **SSLKEYSTOREVALIDITY**: The validity of the keystore, in days. A value of **200** indicates that the validity is 200 days.
prometheus.*	**monitoring**	The details for Prometheus: • **operator.enabled**: Whether to deploy both Service and ServiceMonitor for Prometheus monitoring. The default is **false**. • **operator.interval**: The scrape interval for ServiceMonitor. The default is **15s**. • **operator.scrapeTimeout**: The scrape timeout for ServiceMonitor (should be less than interval). The default is **10s**. • **operator.release**: The helm release name given while deploying Prometheus Operator. The default is **prometheus**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
sdk.*	**job**	The details for SDK: • **name**: The name of the SDK job. The default is **sdk**. • **command**: The command to execute for the SDK job. The default is an empty value. • **runjob**: Whether the SDK job needs to be run as part of the deployment (**true**) or not (**false**). The default is **false**. If set to **true**, a default SDK job is run as part of the Helm installation or upgrade. • **resources.***: The resources for SDK job. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.
customerloader.*	**job**	The details for customerloader: • **name**: The name of the customer loader job. The default is **customerloader**. • **command**: The setting is used to load the product cross-reference data, set -loadCrossRefData <customer_updater_name for required schema>. The default is **-incremental customerupdater1**. • **runjob**: Whether the customer loader job should be run. The default is **false**. • **resources.***: The resources for customer loader job. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
jmsconfig.*	**job**	The details for jmsconfig: • **name**: The name of the JMS configuration job. The default is **jmsconfig**. • **command**: The command to execute for the JMS configuration job. The default is an empty value. • **runjob**: Whether the JMS configuration job should be run. Set to "true" to enable. The default is **false**. • **preCreateJmsServerAndModule**: Whether JMS server and module should be pre-created. Set to "true" to enable. The default is **false**. • **adminServerName**: The admin server name setting specifies the configured value. The default is **Adminserver**. • **resources.***: The resources for JMS configuration job. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
chargingConfiguration Reloader.*	**job**	The details for chargingConfigurationReloader: • **name**: The name of the charging configuration reloader job. The default is **charging-configuration-reloader**. • **reloadAppConfig.runjob**: Whether the application configuration reload job runs. The default is **false**. • **reloadAppConfig.command**: The specify mBean name to be reloaded in AppConfig in below command parameter. for example, command: "charging.server". The default is an empty value. • **reloadLogging.runjob**: Whether the logging reload job runs. The default is **false**. • **reloadLogging.command**: The specify Operation name, Logger or FunctionalDomain and LogLevel in below command parameter. for example, command: "setGridLogLevel oracle.communication.brm.charging.appconfiguration ERROR" Supported Operations are setGridLogLevel, setLogLevel, setGridLogLevelForFunctionalDomain, setLogLevelForFunctionalDomain and updateSubscriberTraceConfiguration. To persist the log level changes, update log4j2.logger.<loggerName> in override or value.yaml. The default is an empty value. • **resources.***: The resource configuration specifies the CPU and memory resources for the component. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
chargingSyCacheMigration.*	**job**	The details for chargingSyCacheMigration: • **name**: The name identifies the configuration entry. The default is **charging-sy-cache-migration**. • **syCacheMigration.runjob**: Whether the Sy cache migration job runs. The default is **false**. • **syCacheMigration.rollback**: Whether the Sy cache migration is rolled back. The default is **false**. • **resources.***: The resource configuration specifies the CPU and memory resources for the component. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.
schemaLoader.*	**job**	The details for schemaLoader: • **resources.***: The minimum and maximum CPU and memory resources for ece-persistence-job and ece-persistence-upgrade-job. See " Setting Minimum and Maximum CPU and Memory Values " in *BRM Cloud Native System Administrator's Guide*. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.
serviceFqdn	**emgateway**	The service FQDN setting specifies the configured value. The default is **ece-emg**.
resources.*	**emgateway**	The resource configuration specifies the CPU and memory resources for the component.
nodeSelector.*	**emgateway**	The node selector configuration specifies where the pod can be scheduled.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
affinity.*	**emgateway**	The details for affinity: • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.weight**: The weight setting specifies the configured value. The default is **100**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.key**: The key setting specifies the configured value. Values vary by list entry. Values in values.yaml include **app, application**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.operator**: The operator setting specifies the configured value. The default is **In**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.values**: The values setting specifies the configured value. The default values are **emgateway**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.topologyKey**: The topology key setting specifies the configured value. The default is **kubernetes.io/hostname**.
addOnPodSpec.*	**emgateway**	The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
emGatewayList.*	**emgateway**	The details for emGatewayList: • **coherenceMemberName**: The first entry in the EM Gateway configuration list Name within Coherence cluster. Values vary by list entry. Values in values.yaml include **emgateway1, emgateway2**. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). • **replicas**: The number of replica instances for this gateway. The default is **1**. • **jmxEnabled**: Whether to enable JMX for monitoring and management. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The counter for the number of times the service has been restarted. The default is **0**. • **emGatewayConfiguration.name**: The name of this EM Gateway configuration. • **emGatewayConfiguration.port**: The port number on which the EM Gateway listens. The default is **15502**. • **emGatewayConfiguration.threadPoolSize**: The size of the thread pool for handling incoming connections. The default is **10**. • **emGatewayConfiguration.sslEnabled**: The setting indicates if SSL is enabled (1) or not (0). The default is **1**. • **emGatewayConfiguration.socketKeepAlive**: The enables socket keep-alive feature (1) or not (0). The default is **1**. • **emGatewayConfiguration.wallet**: The retained per-instance server wallet filename. Rendered configuration uses charging.brmWalletServerLocation. The default is **cwallet.sso**. • **emGatewayConfiguration.clientAuthenticationEnabled**: The setting indicates if client authentication is required (1) or not (0). The default is **0**. • **emGatewayConfiguration.updateResponseTimeout**: The timeout in milliseconds for update responses. The default is **30000**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
wallet	**radiusgateway**	The primary root wallet filename. The chart resolves it under /home/charging/wallet/brmwallet/ root. The default is **cwallet.sso**.
queueSize	**radiusgateway**	The size of the processing queue. The default is **8**.
avpName	**radiusgateway**	The attribute-Value Pair name for RADIUS. The default is **Service-Type**.
vendorId	**radiusgateway**	The vendor ID for RADIUS attributes. The default is **0**.
keyPass	**radiusgateway**	The password for key. The default is an empty value.
keyStoreLocation	**radiusgateway**	The location of the keystore, if SSL is used. The default is an empty value.
enableRetransmissionCheck	**radiusgateway**	Whether to enable checking for retransmissions. The default is **true**.
timeToLive	**radiusgateway**	The time to live for messages in milliseconds. The default is **30000**.
notificationListenerConnectionPoolSize	**radiusgateway**	The connection pool size for notification listeners. The default is **10**.
maxSessionsPerUser	**radiusgateway**	The maximum number of sessions for a user. The default is **5**.
resources.*	**radiusgateway**	The resource configuration specifies the CPU and memory resources for the component.
nodeSelector.*	**radiusgateway**	The node selector configuration specifies where the pod can be scheduled.
affinity.*	**radiusgateway**	The affinity configuration specifies pod scheduling affinity rules.
addOnPodSpec.*	**radiusgateway**	The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
radiusgatewayList.*	**radiusgateway**	The details for radiusgatewayList: • **coherenceMemberName**: The configuration for the first RADIUS Gateway instance Name within Coherence cluster. Values vary by list entry. Values in values.yaml include **radiusgateway1, radiusgateway2**. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **replicas**: The number of replica instances for this gateway. The default is **1**. • **jmxEnabled**: Whether to enable Java Management Extensions for monitoring. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The tracks the number of times the gateway has been restarted. The default is **0**. • **radiusServiceNodePort**: The port on which the Radius service listens for incoming connections. The default is an empty value. • **radiusGatewayConfiguration.name**: The name of this Radius Gateway configuration. Values vary by list entry. Values in values.yaml include **radiusgateway1, radiusgateway2**. • **radiusGatewayConfiguration.radiusTrafficPort**: The port for RADIUS traffic. The default is **1812**. • **radiusGatewayConfiguration.ioThreadPoolSize**: The size of the I/O thread pool. The default is **16**. • **radiusGatewayConfiguration.noOfChallenges**: The number of challenge messages to be sent. The default is **1**. • **radiusGatewayConfiguration.eceWalletLocation**: The ECE wallet location. The default is **/home/charging/wallet/brmwallet/client**. • **radiusGatewayConfiguration.disconnectMessagePort**: The port number of NAS client for sending disconnect messages. The default is **3799**. • **radiusGatewayConfiguration.kafkaBatchSize**: The batch size for Kafka processing. The default is **10**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **radiusGatewayConfiguration.disconnectMessageRetryThreadPoolSize**: The size of the scheduled thread pool executor to resend disconnect messages. The default is **5**. • **radiusGatewayConfiguration.disconnectMessageMaxRetryCount**: The maximum retry count of a disconnect message. The default is **3**. • **radiusGatewayConfiguration.disconnectMessageRetryDelayMillis**: The initial delay after which the retry mechanism is invoked. The default is **3000**. • **radiusGatewayConfiguration.disconnectMessageResponseTimeoutMillis**: The wait time for receiving a disconnect message response from network. The default is **2000**.
resources.*	**diametergateway**	The resource configuration specifies the CPU and memory resources for the component.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
diametergatewayList.*	**diametergateway**	The details for diametergatewayList: • **coherenceMemberName**: The configuration for the first Diameter Gateway instance Name used within Coherence cluster for this member. Values vary by list entry. Values in values.yaml include **diametergateway1, diametergateway2**. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **replicas**: The number of replica instances for this gateway. The default is **1**. • **jmxEnabled**: Whether to enable JMX for monitoring and management. The default is **true**. • **updateStrategyType**: The update Strategy Type for Stateful sets. The default is **OnDelete**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The tracks the number of times the gateway has been restarted. The default is **0**. • **diameterServiceNodePort**: The port on which the Diameter service listens for incoming connections. The default is an empty value. • **nodeSelector.***: The node selector for deployment. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings. • **diameterGatewayConfiguration.name**: The configuration name. Values vary by list entry. Values in values.yaml include **diametergateway1, diametergateway2**. • **diameterGatewayConfiguration.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **diameterGatewayConfiguration.diameterTrafficPort**: The port for Diameter traffic. The default is **3868**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **diameterGatewayConfiguration.ioThreadPoolSize**: The size of the I/O thread pool. The default is **10**. • **diameterGatewayConfiguration.responseTimeout**: The timeout for responses in seconds. The default is **10**. • **diameterGatewayConfiguration.watchDogInterval**: The interval for watchdog checks in seconds. The default is **30**. • **diameterGatewayConfiguration.originHost**: The origin host name for Diameter messages. The default is **ece.example.com**. • **diameterGatewayConfiguration.originRealm**: The origin realm for Diameter messages. The default is **example.com**. • **diameterGatewayConfiguration.diameterTrafficHost**: The specific host for Diameter traffic. The default is an empty value. • **diameterGatewayConfiguration.diameterTrafficHostSctp**: The host for SCTP Diameter traffic. The default is an empty value. • **diameterGatewayConfiguration.loopback**: Whether to use loopback interface, typically for testing. The default is **false**. • **diameterGatewayConfiguration.requestProcessorThreadPoolSize**: The thread pool size for processing requests. The default is **10**. • **diameterGatewayConfiguration.requestProcessorBatchSize**: The batch size for processing requests. The default is **10**. • **diameterGatewayConfiguration.notificationThreadPoolSize**: The thread pool size for notifications. The default is **10**. • **diameterGatewayConfiguration.maxNotificationCommitSize**: The maximum size for committing notifications. The default is **100**. • **diameterGatewayConfiguration.maxWaitTimeNotificationCommit**: The maximum wait time for committing notifications. The default is **3600000**. • **diameterGatewayConfiguration.ccFailover**: The failover support for Credit-Control. The default is **FAILOVER_SUPPORTED**. • **diameterGatewayConfiguration.creditControlFailureHandling**: The handling for Credit-Control failures. The default is **RETRY_AND_TERMINATE**. • **diameterGatewayConfiguration.directDebitingFailureHandling**: The handling for direct debiting failures. The default is **TERMINATE_OR_BUFFER**. • **diameterGatewayConfiguration.sctpMaxInStream**: The maximum inbound streams for SCTP, 0 for default. The default is **0**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **diameterGatewayConfiguration.sctpMaxOutStream**: The maximum outbound streams for SCTP, 0 for default. The default is **0**. • **diameterGatewayConfiguration.sctpSendBufferSize**: The send buffer size for SCTP, 0 for default. The default is **0**. • **diameterGatewayConfiguration.sctpReceiveBufferSize**: The receive buffer size for SCTP, 0 for default. The default is **0**. • **diameterGatewayConfiguration.notificationListenerConnectionPoolSize**: The connection pool size for notification listeners. The default is **10**. • **diameterGatewayConfiguration.allowMultipleConnectionsPerPeer**: Whether to allow multiple connections per peer. Set to "true" to enable. The default is **false**. • **diameterGatewayConfiguration.kafkaPartition**: The specifies the Kafka partition the notification listener should consume messages from. Values vary by list entry. Values in values.yaml include **3, 4**. • **diameterGatewayConfiguration.dgwNotifNetworkResponseTimeoutAdder**: The dgw notif network response timeout adder setting specifies the configured value. The default is **200**. • **diameterGatewayConfiguration.ingressRequestTimeoutMs**: The number of milliseconds beyond the creation time before a request is considered expired during processing. The default is **3000**. • **diameterGatewayConfiguration.enableNetworkExpiration**: Whether network expiration processing uses the incoming timestamp. The default is **true**. • **diameterGatewayConfiguration.rarNotificationAgeLimitInSec**: The time limit (in seconds) after which any Gy notification consumed by Diameter Gateway will be considered expired. The default is **300**. • **diameterGatewayConfiguration.snrNotificationAgeLimitInSec**: The time limit (in seconds) after which any Sy notification consumed by Diameter Gateway will be considered expired. The default is **300**. • **diameterGatewayConfiguration.enrichMSCCForAllRatingGroups**: Whether the enrich MSCC for all rating groups option is enabled. The default is **true**.
resources.*	**cdrgateway**	The resource configuration specifies the CPU and memory resources for the component.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
nodeSelector.*	**cdrgateway**	The node selector configuration specifies where the pod can be scheduled.
affinity.*	**cdrgateway**	The details for affinity: • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.weight**: The weight setting specifies the configured value. The default is **100**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.key**: The key setting specifies the configured value. Values vary by list entry. Values in values.yaml include **app, application**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.operator**: The operator setting specifies the configured value. The default is **In**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.values**: The values setting specifies the configured value. The default values are **cdrgateway**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.topologyKey**: The topology key setting specifies the configured value. The default is **kubernetes.io/hostname**.
addOnPodSpec.*	**cdrgateway**	The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
cdrgatewayList.*	**cdrgateway**	The details for cdrgatewayList: • **coherenceMemberName**: The configuration for the CDR Gateway instance Name used within Coherence cluster for this member. The default is **cdrgateway1**. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **replicas**: The number of replica instances for this gateway. The default is **1**. • **jmxEnabled**: Whether to enable JMX for monitoring and management. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The tracks the number of times the gateway has been restarted. The default is **0**. • **cdrGatewayConfiguration.name**: The configuration name. The default is **cdrgateway1**. • **cdrGatewayConfiguration.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **cdrGatewayConfiguration.primaryInstanceName**: The name of the primary instance for this CDR Gateway. The default is **cdrgateway1**. • **cdrGatewayConfiguration.schemaNumber**: The schema version number. The default is **1**. • **cdrGatewayConfiguration.isNoSQLConnection**: Whether NoSQL connection is used. The default is **true**. • **cdrGatewayConfiguration.noSQLConnectionName**: The name of the NoSQL connection. The default is **noSQLConnection**. • **cdrGatewayConfiguration.connectionName**: The name of the database connection. The default is **oraclePersistence1**. • **cdrGatewayConfiguration.cdrPort**: The port for CDR service. The default is **8084**. • **cdrGatewayConfiguration.cdrHost**: The hostname for the CDR service. The default is **ece-cdrgatewayservice**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **cdrGatewayConfiguration.individualCdr**: Whether to handle CDRs individually. Set to "true" to enable. The default is **false**. • **cdrGatewayConfiguration.cdrServerCorePoolSize**: The core pool size for CDR server thread pool. The default is **32**. • **cdrGatewayConfiguration.cdrServerMaxPoolSize**: The maximum pool size for CDR server thread pool. The default is **256**. • **cdrGatewayConfiguration.enableIncompleteCdrDetection**: Whether incomplete CDR detection is enabled. Set to "true" to enable. The default is **false**. • **cdrGatewayConfiguration.retransmissionDuplicateDetectionEnabled**: Whether duplicate CDR detection for retransmissions is enabled. Set to "true" to enable. The default is **false**.
serverSslKeyStore	**httpgateway**	The SSL keystore file for the server. The default is **httpGatewayServer.jks**.
extServerSSLKeyStoreSecret	**httpgateway**	The names of the external Kubernetes secrets for the HTTP Gateway. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information.
serverSslKeyStoreType	**httpgateway**	The type of the SSL keystore. The default is **PKCS12**.
serverSslKeyStoreAlias	**httpgateway**	The alias for the key in the SSL keystore. The default is **oracle**.
walletLocation	**httpgateway**	The wallet location for SSL configuration. The default is **/home/charging/wallet/ecewallet/**.
snrHttp2Enable	**httpgateway**	Whether to enable HTTP2 Server Name Indication. The default is **true**.
cdrGenerationEnabled	**httpgateway**	Whether CDR generation is enabled. Set to "true" to enable. The default is **false**.
cdrGenerationStandaloneMode	**httpgateway**	Whether CDR generation uses standalone mode. The default is **false**.
cdrGatewayList	**httpgateway**	The list of CDR Gateway services. The default is **ece-cdrgatewayservice:8084**.
cdrGatewayRetry	**httpgateway**	The number of retries for CDR Gateway communication. The default is **3**.
retryIntervalInMillis	**httpgateway**	The interval between retries in milliseconds. The default is **5000**.
serverHttpExternalPort	**httpgateway**	The external HTTP port for the server. The default is **31500**.
rateOfflineCDRinRealtime	**httpgateway**	Whether to rate offline CDRs in real-time. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
generateCDRsForOnlineRequests	**httpgateway**	Whether to generate CDRs for online requests. The default is **true**.
callScreeningEnabled	**httpgateway**	Whether to enable call screening feature. The default is **true**.
notificationListenerConnectionPoolSize	**httpgateway**	The connection pool size for notification listeners. The default is **10**.
ecsClusterHealthStatus	**httpgateway**	The ECS cluster health status value used for active-active site health detection. The default value is an empty string. This is a read-only value that is automatically updated by the HTTP Gateway.
httpIdentityKeystore	**httpgateway**	The SSL identity keystore file for the server. The default is an empty value.
extHttpIdentityKeystoreSecret	**httpgateway**	The names of the external Kubernetes secrets for the HTTP Gateway. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information.
httpIdentityKeystoreType	**httpgateway**	The type of the SSL identity keystore. The default is an empty value.
httpTruststore	**httpgateway**	The SSL truststore file. The default is an empty value.
extHttpTruststoreSecret	**httpgateway**	The names of the external Kubernetes secrets for the HTTP Gateway. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information.
httpTruststoreType	**httpgateway**	The type of the SSL truststore. The default is an empty value.
nrfPublicKeyLocation	**httpgateway**	The public Key Path of NRF so that JWT can be validated. The default is an empty value.
extNrfPublicKeyLocationSecret	**httpgateway**	The names of the external Kubernetes secrets for the HTTP Gateway. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information.
nrfJwtAlgorithm	**httpgateway**	The algorithm used to create the NRF Public Key. Valid Values: RSA,EC. Default value is RSA. The default is **RSA**.
resources.*	**httpgateway**	The resource configuration specifies the CPU and memory resources for the component.
nodeSelector.*	**httpgateway**	The node selector configuration specifies where the pod can be scheduled.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
affinity.*	**httpgateway**	The details for affinity: • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.weight**: The weight setting specifies the configured value. The default is **100**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.key**: The key setting specifies the configured value. Values vary by list entry. Values in values.yaml include **app, application**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.operator**: The operator setting specifies the configured value. The default is **In**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.values**: The values setting specifies the configured value. The default values are **ece-httpgateway**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.topologyKey**: The topology key setting specifies the configured value. The default is **kubernetes.io/hostname**.
addOnPodSpec.*	**httpgateway**	The additional pod specification defines extra pod-level settings.
includeTrailingForwardSlash	**httpgateway**	Whether a trailing forward slash is included. The default is **true**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
httpgatewayList.*	**httpgateway**	The details for httpgatewayList: • **coherenceMemberName**: The configuration for the first HTTP Gateway instance Name used within Coherence cluster for this member. Values vary by list entry. Values in values.yaml include **httpgateway1, httpgateway2**. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **replicas**: The number of replica instances for this gateway. The default is **1**. • **maxreplicas**: The maximum number of replicas for auto-scaling. The default is **3**. • **averageCpuUtilization**: The CPU utilization threshold for auto-scaling. The default is **70**. • **averageMemoryUtilization**: The memory utilization threshold for auto-scaling. The default is an empty value. • **scaleDownStabilizationWindowSeconds**: The stabilization window before scaling down. The default is **120**. • **disableHpaScaleDown**: Whether HPA scale-down is disabled. The default is **false**. • **jmxEnabled**: Whether to enable Java Management Extensions for monitoring. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The tracks the number of times the gateway has been restarted. The default is **0**. • **httpGatewayConfiguration.name**: The configuration name. Values vary by list entry. Values in values.yaml include **httpgateway1, httpgateway2**. • **httpGatewayConfiguration.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **httpGatewayConfiguration.serverHttp2Enabled**: Whether to enable HTTP/2 server. The default is **true**. • **httpGatewayConfiguration.serverPort**: The secure server port for HTTPS. The default is **8443**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **httpGatewayConfiguration.serverHttpPort**: The server port for HTTP. The default is **8080**. • **httpGatewayConfiguration.nrfHeartBeatRetryCount**: The number of retries for NRF heartbeats. The default is **4**. • **httpGatewayConfiguration.serverSslEnabled**: Whether SSL is enabled for the server. Set to "true" to enable. The default is **false**. • **httpGatewayConfiguration.processingThreadPoolSize**: The size of the processing thread pool. The default is **200**. • **httpGatewayConfiguration.processingQueueSize**: The size of the processing queue. The default is **32768**. • **httpGatewayConfiguration.kafkaBatchSize**: The batch size for Kafka processing. The default is **10**. • **httpGatewayConfiguration.externalTrafficInfo**: The external traffic information. The default is an empty value. • **httpGatewayConfiguration.nrfRetryIntervalInSecond**: The interval (in seconds) between NRF retries. The default is **60**. • **httpGatewayConfiguration.nrfRetryCount**: The number of retries for NRF requests. The default is **3**. • **httpGatewayConfiguration.kafkaPartition**: The number of Kafka partitions used. Values vary by list entry. Values in values.yaml include **5, 6**. • **httpGatewayConfiguration.scpAuthorities**: The SCP authorities, if applicable. The default is an empty value. • **httpGatewayConfiguration.pcfSSLEnabled**: Whether SSL is enabled for the PCF. Set to "true" to enable. The default is **false**. • **httpGatewayConfiguration.nrfSSLEnabled**: Whether SSL is enabled for the NRF. Set to "true" to enable. The default is **false**. • **httpGatewayConfiguration.smfSSLEnabled**: Whether SSL is enabled for the SMF. Set to "true" to enable. The default is **false**. • **httpGatewayConfiguration.httpSSLType**: The type of SSL configuration (Set to "twoway" for mutual SSL, or "oneway"). The default is an empty value. • **httpGatewayConfiguration.oauth2Enabled**: Whether to allow OAuth2 support. The default is **false**. • **httpGatewayConfiguration.ingressRequestTimeoutMs**: The number of milliseconds beyond the creation time before a request is

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		considered expired during processing. The default is **3000**. • **httpGatewayConfiguration.enableNetworkExpiration**: Whether network expiration processing uses the incoming timestamp. The default is **true**. • **httpGatewayConfiguration.chfNetworkId**: The CHF Network ID for the HTTP Gateway. The default is an empty value. • **httpGatewayConfiguration.ccnNotificationAgeLimitInSec**: The time limit (in seconds) after which any CCN notification consumed by HTTP Gateway will be considered expired. The default is **300**. • **httpGatewayConfiguration.slcnNotificationAgeLimitInSec**: The time limit (in seconds) after which any SLCN notification consumed by HTTP Gateway will be considered expired. The default is **300**. • **httpGatewayConfiguration.managementServerPort**: The management server port setting specifies the configured value. The default is **9000**. • **httpGatewayConfiguration.serverUndertowWorkerThreads**: The server undertow worker threads setting specifies the configured value. The default is **32**. • **httpGatewayConfiguration.http2MaxConcurrentStreams**: The http2 max concurrent streams setting specifies the configured value. The default is **-1**. • **httpGatewayConfiguration.http2InitialWindowSize**: The http2 initial window size setting specifies the configured value. The default is **-1**.
resources.*	**customerUpdater**	The resource configuration specifies the CPU and memory resources for the component.
nodeSelector.*	**customerUpdater**	The node selector configuration specifies where the pod can be scheduled.
affinity.*	**customerUpdater**	The affinity configuration specifies pod scheduling affinity rules.
addOnPodSpec.*	**customerUpdater**	The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
customerUpdaterList.*	**customerUpdater**	The details for customerUpdaterList: • **schemaNumber**: The configuration for the Customer Updater service Schema number for database interaction. The default is **1**. • **coherenceMemberName**: The coherence cluster member name. The default is **customerupdater1**. • **replicas**: The number of replica instances. The default is **1**. • **jmxEnabled**: Whether to enable Java Management Extensions for monitoring. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). The default is an empty value. • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **restartCount**: The tracks the number of times this customerUpdater has been restarted. The default is **0**. • **oracleQueueConnectionConfiguration.name**: The configuration name. The default is **customerupdater1**. • **oracleQueueConnectionConfiguration.gatewayName**: The gateway name for database connection. The default is **customerupdater1**. • **oracleQueueConnectionConfiguration.hostName**: The database host name. The default is an empty value. • **oracleQueueConnectionConfiguration.port**: The database port. The default is **1521**. • **oracleQueueConnectionConfiguration.sid**: The if CN Database service doesn't support sid, then set sid as "" and provide correct jdbc url instead. The default is **pindb**. • **oracleQueueConnectionConfiguration.userName**: The database user name. The default is **pin**. • **oracleQueueConnectionConfiguration.jdbcUrl**: The JDBC URL for database connection, if SID not used. The default is an empty value. • **oracleQueueConnectionConfiguration.queueName**: The queue name for sync

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		operations. The default is **IFW_SYNC_QUEUE**. • **oracleQueueConnectionConfiguration.suspenseQueueName**: The queue name for suspense operations. The default is **ECE_SUSPENSE_QUEUE**. • **oracleQueueConnectionConfiguration.ackQueueName**: The queue name for acknowledgment operations. The default is **ECE_ACK_QUEUE**. • **oracleQueueConnectionConfiguration.amtAckQueueName**: The queue name for amount acknowledgment operations. The default is **ECE_AMT_ACK_QUEUE**. • **oracleQueueConnectionConfiguration.dbSSLEnabled**: Whether SSL is enabled for the database connection. Set to "true" to enable. The default is **false**. • **oracleQueueConnectionConfiguration.tlsVersion**: The TLS version to support, such as 1.2 or 1.3 • **oracleQueueConnectionConfiguration.dbSSLType**: The type of database SSL configuration (e.g., "twoway" for mutual SSL). The default is **oneway**. • **oracleQueueConnectionConfiguration.sslServerCertDN**: The distinguished Name (DN) of the server SSL certificate for database connection. The default is **DC=local,DC=oracle,CN=pindb**. • **oracleQueueConnectionConfiguration.trustStoreLocation**: The trust store location for SSL. The default is **cwallet.sso**. • **oracleQueueConnectionConfiguration.trustStoreType**: The trust store type. The default is **SSO**. • **oracleQueueConnectionConfiguration.batchSize**: The batch size for processing. The default is **1**. • **oracleQueueConnectionConfiguration.dbTimeout**: The database timeout in seconds. The default is **900**. • **oracleQueueConnectionConfiguration.retryCount**: The number of retries for database operations. The default is **10**. • **oracleQueueConnectionConfiguration.retryInterval**: The interval between retries in seconds. The default is **60**. • **oracleQueueConnectionConfiguration.walletLocation**: The wallet location for SSL configuration. The default is **/home/charging/wallet/ecewallet/**. • **oracleQueueConnectionConfiguration.oraFilesLocation**: The location of Oracle files used

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		for database connections or configurations (e.g., Oracle Database Resource Manager files). The default is **/home/charging/ext/ora_files/brm/**. • **oracleQueueConnectionConfiguration.extBRMDBSSLWalletSecret**: The name of the external Kubernetes secret containing the SSL TrustStore. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information.
resources.*	**ratedEventFormatter**	The resource configuration specifies the CPU and memory resources for the component.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
ratedEventFormatterList.*	**ratedEventFormatter**	The details for ratedEventFormatterList: • **schemaNumber**: The configuration for the Rated Event Formatter service Schema number for database interaction. The default is **1**. • **replicas**: The number of replica instances. The default is **1**. • **coherenceMemberName**: The coherence cluster member name. The default is **ratedeventformatter1**. • **jmxEnabled**: Whether the component is JMX-enabled (**true**) or not (**false**). In an active-active deployment, set the **jmxEnabled** configuration entry to **true** for every primary and secondary Rated Event Formatter pod. For configuration details, see "Setting Up Active-Active Disaster Recovery for ECE" in *BRM Cloud Native System Administrator's Guide*. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **restartCount**: The tracks the number of times this ratedEventFormatter has been restarted. The default is **0**. • **nodeSelector.***: The node selector for deployment. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings. • **ratedEventFormatterConfiguration.name**: The configuration name. The default is **ratedeventformatter1**. • **ratedEventFormatterConfiguration.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **ratedEventFormatterConfiguration.primaryInstanceName**: The name of the primary instance. The default is **ratedeventformatter1**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **ratedEventFormatterConfiguration.partition** : The partition number for data segmentation. The default is **1**. • **ratedEventFormatterConfiguration.noSQLConnectionName**: The name of the NoSQL connection. The default is **noSQLConnection**. • **ratedEventFormatterConfiguration.connectionName**: The name of the database connection. The default is **oraclePersistence1**. • **ratedEventFormatterConfiguration.threadPoolSize**: The size of the thread pool for processing. The default is **6**. • **ratedEventFormatterConfiguration.retainDuration**: The duration to retain data, 0 for indefinitely. The default is **0**. • **ratedEventFormatterConfiguration.ripeDuration**: The duration until data is considered ripe for processing in seconds. The default is **600**. • **ratedEventFormatterConfiguration.checkPointInterval**: The interval for checkpointing in seconds. The default is **6**. • **ratedEventFormatterConfiguration.maxPersistenceCatchupTime**: The maximum time for persistence catch-up, 0 for no limit. The default is **0**. • **ratedEventFormatterConfiguration.siteName**: The parameter siteName, Each REF instance is supposed to handle the rated events for, in ECE active-active setup. The siteName is to determine which RatedEvent DB table to use. For the primary REF running in a site1 (BRM) and secondary REF instances running the remote sites (BRM2), siteName parameter value is BRM and RatedEvent DB table to use is RatedEvent_BRM. The default is an empty value. • **ratedEventFormatterConfiguration.pluginPath**: The use remCdrPlugin and jar to integrate with REM. The default is **brm-rated-event-manager-brmJarsVersion.jar**. • **ratedEventFormatterConfiguration.pluginType**: The plugin type setting specifies the configured value. The default is **com.oracle.brm.ref_brm_plugin.RatedEventManagerCdrPlugin**. • **ratedEventFormatterConfiguration.pluginName**: The plugin name setting specifies the configured value. The default is **brmCdrPlugin1**. • **ratedEventFormatterConfiguration.noSQLBatchSize**: The no SQL batch size setting

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		specifies the configured value. The default is **25**.
resources.*	**cdrFormatter**	The resource configuration specifies the CPU and memory resources for the component.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
cdrFormatterList.*	**cdrFormatter**	The details for cdrFormatterList: • **schemaNumber**: The configuration for the cdr Formatter service Schema number for database interaction. The default is **1**. • **replicas**: The number of replica instances. The default is **1**. • **coherenceMemberName**: The coherence cluster member name. The default is **cdrformatter1**. • **jmxEnabled**: Whether to enable Java Management Extensions for monitoring. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **restartCount**: The tracks the number of times this ratedEventFormatter has been restarted. The default is **0**. • **nodeSelector.***: The node selector for deployment. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings. • **cdrFormatterConfiguration.name**: The configuration name. The default is **cdrformatter1**. • **cdrFormatterConfiguration.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **cdrFormatterConfiguration.primaryInstanceName**: The name of the primary instance. The default is **cdrformatter1**. • **cdrFormatterConfiguration.schemaNumber** : The schema number for database interaction. The default is **1**. • **cdrFormatterConfiguration.isNoSQLConnection**: Whether NoSQL connection is used. The default is **true**. • **cdrFormatterConfiguration.noSQLConnectionName**: The name of the NoSQL connection. The default is **noSQLConnection**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **cdrFormatterConfiguration.connectionName**: The name of the database connection. The default is **oraclePersistence1**. • **cdrFormatterConfiguration.threadPoolSize**: The size of the thread pool for processing. The default is **6**. • **cdrFormatterConfiguration.retainDuration**: The duration to retain data, 0 for indefinitely. The default is **0**. • **cdrFormatterConfiguration.ripeDuration**: The duration until data is considered ripe for processing in seconds. The default is **60**. • **cdrFormatterConfiguration.checkPointInterval**: The interval for checkpointing in seconds. The default is **6**. • **cdrFormatterConfiguration.maxPersistenceCatchupTime**: The maximum time for persistence catch-up, 0 for no limit. The default is **0**. • **cdrFormatterConfiguration.pluginPath**: The setting specifies the path to a Java Archive (JAR) file. The default is **ece-cdrformatter.jar**. • **cdrFormatterConfiguration.pluginType**: The setting indicates the fully qualified class name of the plugin implementation. The default is **oracle.communication.brm.charging.cdr.formatterplugin.internal.SampleCdrFormatterCustomPlugin**. • **cdrFormatterConfiguration.pluginName**: The plugin name setting specifies the configured value. The default is **cdrFormatterPlugin1**. • **cdrFormatterConfiguration.noSQLBatchSize**: The no SQL batch size setting specifies the configured value. The default is **25**. • **cdrFormatterConfiguration.cdrStoreFetchSize**: The setting defines the fetch size for retrieving CDRs (Call Detail Records) from storage. The default is **2500**. • **cdrFormatterConfiguration.cdrOrphanRecordCleanupAgeInSec**: The setting specifies the age (in seconds) at which orphaned CDR records should be cleaned up. The default is **200**. • **cdrFormatterConfiguration.cdrOrphanRecordCleanupSleepIntervalInSec**: The determines the sleep interval (in seconds) for the cleanup process mentioned above. The default is **200**. • **cdrFormatterConfiguration.enableIncompleteCdrDetection**: Whether incomplete CDR detection is enabled or not. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
resources.*	**brmGateway**	The resource configuration specifies the CPU and memory resources for the component.
nodeSelector.*	**brmGateway**	The node selector configuration specifies where the pod can be scheduled.
affinity.*	**brmGateway**	The details for affinity: • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.weight**: The weight setting specifies the configured value. The default is **100**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.key**: The key setting specifies the configured value. Values vary by list entry. Values in values.yaml include **app, application**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.operator**: The operator setting specifies the configured value. The default is **In**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.values**: The values setting specifies the configured value. The default values are **brmgateway**. • **podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.topologyKey**: The topology key setting specifies the configured value. The default is **kubernetes.io/hostname**.
addOnPodSpec.*	**brmGateway**	The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
brmGatewayList.*	**brmGateway**	The details for brmGatewayList: • **schemaNumber**: The configuration for the brmgateway instance Schema number for database interaction. The default is **1**. • **replicas**: The number of replica instances. The default is **1**. • **coherenceMemberName**: The coherence cluster member name. The default is **brmgateway1**. • **jmxEnabled**: Whether to enable Java Management Extensions for monitoring. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The tracks the number of times this brmGateway has been restarted. The default is **0**. • **jmxport**: The JMX port for Java Management Extensions (left blank, indicating default or not used). The default is an empty value. • **brmGatewayConfiguration.name**: The configuration name. The default is **brmgateway1**. • **brmGatewayConfiguration.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **brmGatewayConfiguration.emptyQueueThreadSleepInterval**: The setting specifies the sleep interval (in milliseconds) for a thread responsible for handling empty queues. The default is **50**. • **brmGatewayConfiguration.jmsReceiveSleepInterval**: The setting sets the sleep interval (in milliseconds) for a JMS (Java Message Service) receive operation. The default is **100**. • **brmGatewayConfiguration.brmResponseTimeOutInterval**: The timeout interval (in milliseconds) for receiving a response from a BRM (Billing and Revenue Management) system. The default is **600000**. • **brmGatewayConfiguration.gatewaySleepInterval**: The gateway sleep interval setting specifies the configured value. The default is **2000**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **brmGatewayConfiguration.jmsBatchSize**: The JMS batch size setting specifies the configured value. The default is **10**. • **brmGatewayConfiguration.jmsReceiveTimeout**: The JMS receive timeout setting specifies the configured value. The default is **2000**. • **brmGatewayConfiguration.brmWorkerThreads**: The setting indicates the number of worker threads allocated for handling tasks related to BRM (Billing and Revenue Management) operations. The default is **10**. • **brmGatewayConfiguration.brmSchedulerThreadInitialDelay**: The setting specifies the initial delay before starting a scheduler thread related to BRM operations. The default is **10**. • **brmGatewayConfiguration.brmSchedulerThreadDelayPeriod**: The setting specifies the delay period between successive executions of the scheduler thread for BRM operations. The default is **3**. • **brmGatewayConfiguration.brmSuspenseQueuePeriod**: The setting defines the period after which items in the suspense queue related to BRM operations are processed. The default is **1800000**. • **brmGatewayConfiguration.connectionRetryCount**: The setting specifies the number of retry attempts for establishing a connection. If a connection attempt fails, the system may retry up to this number of times. The default is **10**. • **brmGatewayConfiguration.connectionRetryInterval**: The setting specifies the interval between retry attempts for establishing a connection. The default is **10000**. • **brmGatewayConfiguration.kafkaPartition**: The number of Kafka partitions used. The default is **1**.
resources.*	**monitoringAgent**	The resource configuration specifies the CPU and memory resources for the component.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
monitoringAgentList.*	**monitoringAgent**	The details for monitoringAgentList: • **name**: The monitoring-agent JMX servers remain unauthenticated and non-TLS for federation compatibility. Configuration name. Values vary by list entry. Values in values.yaml include **monitoringagent1, monitoringagent2**. • **replicas**: The number of replica instances. The default is **1**. • **jmxport**: The JMX port for Java Management Extensions. The default is an empty value. • **jmxEnabled**: Whether to enable Java Management Extensions for monitoring. The default is **true**. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmOpts**: The additional JVM options. • **jvmGCOpts**: The JVM garbage collection options. • **restartCount**: The tracks the number of times this monitoringAgent has been restarted. The default is **0**. • **nodeSelector.***: The node selector for deployment. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.
labels	**charging**	The label for all pods in the deployment. The default value is **ece**.
jmxport	**charging**	The JMX port exposed by ece, which can be used to log in to JConsole. The default is **31022**.
coherencePort	**charging**	The Coherence cluster communication port.
terminationGracePeriodSeconds	**charging**	The time to wait before forcefully terminating the pod. The default is **10**.
persistenceEnabled	**charging**	Whether to persist the ECE cache data into the Oracle database. The default is **true**. See " Enabling Persistence in ECE " in *BRM Cloud Native System Administrator's Guide* for more information.
ratedEventStreamingEnabled	**charging**	Whether to enable rated events to be published to a kafka topic. The default is **false**.
rebuildCdrStoreTable	**charging**	Whether to enable rebuilding of the cdrStore table. The default is **false**.
hpaEnabled	**charging**	Whether to enable autoscaling using Kubernetes Horizontal Pod Autoscaler. The default is **false**. See " Setting Up Autoscaling of ECE Pods " in *BRM Cloud Native System Administrator's Guide* for more information.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
remEnabled	**charging**	Whether to enable Rated Event Manager. The default is **true**.
incrementalCustomerLoad	**charging**	Whether subscribers are loaded incrementally into the ECE cache after the initial customer load. Set this to **true** and configure **migration.loader.initialCustomerLoadFilterQuery** to load the initial set of subscribers during a fresh installation. Configure **migration.loader.incrementalCustomerLoadFilterQuery** to load additional subscribers by using the Customer Loader job during a Helm upgrade after ECE enters the usage processing state. Changing this value after ECE starts can cause the customerUpdater pods to restart. The default is **false**.
parallelPodManagement	**charging**	Whether pods are started, updated, or terminated in parallel (**true**) instead of sequentially (**false**).
timeoutSurvivorQuorum	**charging**	The minimum number of cluster members that must remain in the cluster when the cluster service is terminating suspect members, without data loss. The default is **3**. To calculate the minimum number, use this formula: (**chargingServerWorkerNodes** – 1) * (sum of all ecs pods/**chargingServerWorkerNodes**)
chargingServerWorkerNodes	**charging**	The number of charging server worker nodes. The default is **3**.
clusterDomain	**charging**	The DNS domain configured for the Kubernetes cluster. An empty value uses the cluster's default DNS domain.
autoGenerateAppConfig	**charging**	Whether selected application configurations are automatically replicated from the primary site to all sites in an active-active deployment. When set to true , ECE replicates diameterGatewayConfig , cdrFormatterConfiguration , cdrGatewayConfiguration , httpGatewayConfig , nfProfileConfig , nfServiceConfig , and brmGatewayConfiguration . It does not replicate ratedEventFormatterConfiguration , oraclePersistence , cachePersistence , ratedEventPublisher , Kafka configuration, JMSConfiguration , siteConfig , or customerGroupConfig . You must define these configurations for each site in the override file. When set to false , you must configure all pod and application configurations for every site and ensure that Coherence members are unique in each site. This parameter is intended primarily for large systems. The default is **true**.
metrics.*	**charging**	The details for metrics: • **port**: The port setting specifies the service port. The default is **19612**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
ecs.*	**charging**	The details for ecs: • **jmxport**: The JMX port setting specifies the port used for JMX access. The default is an empty value. • **replicas**: The out of total configured replicas of ecs statefulset, first few replica(s) will be management enabled. value of managementEnabledReplicas attribute must be at least 1 and less than replicas attribute of ecs. The default is **3**. • **maxreplicas**: The maxreplicas setting specifies the configured value. The default is **3**. • **managementEnabledReplicas**: The management enabled replicas setting specifies the configured value. The default is **1**. • **averageCpuUtilization**: The target CPU utilization percentage for autoscaling. The default is **70**. • **averageMemoryUtilization**: The memory utilization threshold for auto-scaling. The default is an empty value. • **resources.***: The resource configuration specifies the CPU and memory resources for the component. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.weight**: The weight setting specifies the configured value. The default is **100**. • **affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.key** : The key setting specifies the configured value. Values vary by list entry. Values in values.yaml include **app, dr**. • **affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.operator**: The operator setting specifies the configured value. The default is **In**. • **affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.labelSelector.matchExpressions.values**: The values setting specifies the configured value. The default values are **ece-federation**. • **affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution.podAffinityTerm.topologyKey**: The topology key setting

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		specifies the configured value. The default is **kubernetes.io/hostname**. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings. • **scaleDownStabilizationWindowSeconds**: The window before scaling down. The default is **120**. • **disableHpaScaleDown**: Whether HPA scale-down is disabled. The default is **false**. • **coherenceMemberName**: The name for coherence member. The default is **ecs**. • **jmxEnabled**: Whether to enable JMX. The default is **true**. • **coherencePort**: The coherence cluster communication port (left blank, indicating default or not used). • **enableCoherenceReporting**: Whether coherence reporting is enabled. The default is **false**. • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The tracks the number of times this has been restarted. The default is **0**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
pricingupdater.*	**charging**	The configuration includes the corresponding component settings. • **jmxport**: The JMX port setting specifies the port used for JMX access. The default is an empty value. • **replicas**: The replicas setting specifies the configured value. The default is **1**. • **coherenceMemberName**: The coherence member name setting specifies the configured value. The default is **pricingupdater**. • **jmxEnabled**: Whether JMX is enabled. The default is **true**. • **coherencePort**: The Coherence port setting specifies the cluster communication port. • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **restartCount**: The restart count records the number of component restarts. The default is **0**. • **resources.***: The resource configuration specifies the CPU and memory resources for the component. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
configLoader.*	**charging**	The details for configLoader: • **coherenceMemberName**: The coherence member name setting specifies the configured value. The default is **configloader**. • **jmxEnabled**: Whether JMX is enabled. The default is **false**. • **jvmGCOpts**: The JVM garbage collection options. • **jvmJMXOpts**: The JVM options for JMX configuration. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **coherencePort**: The Coherence port setting specifies the cluster communication port. • **resources.***: The resource configuration specifies the CPU and memory resources for the component. • **nodeSelector.***: The node selector configuration specifies where the pod can be scheduled. • **affinity.***: The affinity configuration specifies pod scheduling affinity rules. • **addOnPodSpec.***: The additional pod specification defines extra pod-level settings.
customerLoader.*	**charging**	The details for customerLoader: • **coherenceMemberName**: The coherence member name setting specifies the configured value. The default is **customerloader**. • **jvmGCOpts**: The JVM garbage collection options. • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options. • **coherencePort**: The Coherence port setting specifies the cluster communication port.
query.*	**charging**	The details for query: • **jvmCoherenceOpts**: The JVM options for Coherence configuration. • **jvmOpts**: The additional JVM options.
clusterName	**charging**	The clusterName determines site where ref pod is running. The default is **BRM**.
isFederation	**charging**	Whether federation is enabled. The default is **false**.
activeCluster	**charging**	Whether the cluster is marked as primary in an active-active setup. The default is **true**.
secondaryCluster	**charging**	Whether the cluster is configured as a secondary cluster. The default is **false**.
clusterTopology	**charging**	The topology type of the cluster. The default is **active-active**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
cluster.*	**charging**	The details for cluster: • **primary.clusterName**: The name of the primary cluster. • **primary.eceServiceName**: The ECE service name that creates the Kubernetes cluster with all of the ECE components in the primary cluster. The default is **ece-server**. • **primary.eceServicefqdnOrExternalIP**: The fully qualified domain name (FQDN) or external IP address of the ECE service running in the primary cluster. For example: **ece-server.***NameSpace***.svc.cluster.local**. • **secondary.clusterName**: The name of the secondary cluster. • **secondary.eceServiceName**: The ECE service name setting specifies the configured value. The default is an empty value. • **secondary.eceServicefqdnOrExternalIP**: The fully qualified domain name (FQDN) or external IP address of the ECE service running in the secondary cluster. For example: **ece-server.***NameSpace***.svc.cluster.local**.
journalingConfig.*	**charging**	The details for journalingConfig: • **journalManagerDirectoryPath**: The journal manager directory path setting specifies the configured value. The default is **/tmp**. • **flashJournalMaxSize**: The flash journal maximum size setting specifies the configured value. The default is **10GB**. • **ramJournalMaxSize**: The RAM journal maximum size setting specifies the configured value. The default is **10%**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
federatedCacheScheme.*	**charging**	The details for federatedCacheScheme: • **threadCount.brmFederated**: The BRM federated setting specifies the configured value. The default is **32**. • **threadCount.xrefFederated**: The xref federated setting specifies the configured value. The default is **8**. • **threadCount.replicatedFederated**: The replicated federated setting specifies the configured value. The default is **1**. • **threadCount.offerProfileFederated**: The offer profile federated setting specifies the configured value. The default is **1**. • **journalCacheHighUnits.brmJournalCacheHighUnits**: The BRM journal cache high units setting specifies the configured value. The default is **8GB**. • **journalCacheHighUnits.xreffederatedJournalCacheHighUnits**: The xreffederated journal cache high units setting specifies the configured value. The default is **500MB**. • **journalCacheHighUnits.replicatedfederatedJournalCacheHighUnits**: The replicatedfederated journal cache high units setting specifies the configured value. The default is **500MB**. • **journalCacheHighUnits.offerProfileFederatedJournalCacheHighUnits**: The offer profile federated journal cache high units setting specifies the configured value. The default is **500MB**. • **federationPort.brmfederated**: The federation port for BRM. The default is **31016**. • **federationPort.xreffederated**: The federation port for XREF. The default is **31017**. • **federationPort.replicatedfederated**: The unique identifier for the federated replication setup. The default is **31018**. • **federationPort.offerProfileFederated**: The unique identifier for federated offer profiles. The default is **31019**.
eceWalletLocation	**charging**	The path to the ECE wallet directory. The default is **/home/charging/wallet/ecewallet/**.
extBRMClientWalletSecret	**charging**	The existing Kubernetes secret containing all BRM client wallet files. Leave empty to use **secrets/brmwallet/client/**.
extBRMServerWalletSecret	**charging**	The existing Kubernetes secret containing all BRM server wallet files. Leave empty to use **secrets/brmwallet/server/**.
extBRMRootWalletSecret	**charging**	The existing Kubernetes secret containing all BRM root wallet files. Leave empty to use **secrets/brmwallet/root/**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
brmWalletServerLocation	**charging**	The primary BRM server wallet filename. The chart resolves it under **/home/charging/wallet/brmwallet/server**. The default is **cwallet.sso**.
brmWalletClientLocation	**charging**	The retained client wallet filename. Current rendered configuration does not consume it. The default is **cwallet.sso**.
brmWalletLocation	**charging**	The path to the BRM wallet directory. The default is **/home/charging/wallet/brmwallet**.
customSSLWallet	**charging**	Whether a custom SSL wallet is used. The default is **false**.
secretCustomWallet.*	**charging**	The details for secretCustomWallet: • **name**: The name identifies the configuration entry.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
server.*	**charging**	The details for server: • **aopEnabled**: Whether Advanced Overcharging Protection (AOP) is enabled. The default is **false**. • **aopVariance**: The allowed time variance for AOP calculations, expressed as an ISO 8601 duration. The default is **PT10M** (10 minutes). • **degradedModeThreshold**: The threshold at which the system enters degraded mode. The default is **0**. • **reverseRateUseAllBalances**: Whether reverse-rating calculations use all available balances. The default is **false**. • **debitRefundSessionEvictionSize**: The maximum number of debit refund sessions maintained in the eviction cache. The default is **10**. • **currencyScale**: The number of decimal places retained for currency values. The default is **2**. • **currencyRoundingMode**: The rounding mode used for currency calculations. The default is **HALF_UP**. • **nonCurrencyScale**: The number of decimal places retained for non-currency values. The default is **2**. • **nonCurrencyRoundingMode**: The rounding mode used for non-currency calculations. The default is **HALF_UP**. • **nonLinearRatingEnabled**: Whether non-linear rating is enabled. The default is **true**. • **tariffTimeChangeSupported**: Whether tariff time changes are supported. The default is **true**. • **systemConsumptionRule**: The rule used to determine the order in which system resources are consumed. The default is **EARLIEST_START_EARLIEST_EXPIRATION**. • **thresholdBreachNotificationMode**: The operation stage at which threshold-breach notifications are generated. The default is **ON_TERMINATE**. • **accountingOnOffMode**: The action applied when accounting is turned off. The default is **CANCEL**. • **offerEligibilitySelectionMode**: The offer attribute used to determine offer eligibility. The default is **END_TIME**. • **offerSelectionModeOnEquiPriorityOffers**: The offer attribute used to select among offers with equal priority. The default is **START_TIME**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **contractProductPriority**: The contract winner for same-account overlapping Contract overrides. The default is **LATEST_START_DATE**. • **contractRatingEnabled**: Whether to enable contract price and profile selection. The default is **false**. • **alignRecurringImpactsToOffer**: Whether recurring impacts are aligned with the offer start time. The default is **false**. • **sharingRetryCount**: The number of retry attempts performed for sharing operations. The default is **1**. • **remainingBalanceCalcMode**: The mode used to calculate the remaining balance. The default is **NONE**. • **virtualTime**: The virtual date and time used for testing. An empty value means that no explicit virtual time is configured. The default is **an empty value**. • **virtualTimeMode**: The mode controlling virtual-time behavior. The default is **0**. • **concurrentRatedEventFederatedSleepTime** : The sleep interval between concurrent rated events in federated mode. The default is **0**. • **systemCurrencyNumericCode**: The ISO 4217 numeric code of the system currency. The default is **840** (US dollar). • **treatNoRatingGraphAsError**: Whether the absence of a rating graph is treated as an error. The default is **true**. • **matchFactorEnabled**: Whether match-factor calculations are enabled. The default is **false**. • **skipCreditFloorBreachCheck**: Whether credit-floor breach checks are skipped. The default is **false**. • **rateZeroQuantity**: Whether events with a quantity of zero are rated. The default is **false**. • **groupNotificationEnabled**: Whether group notifications are enabled. The default is **false**. • **brmPostCommitEnabled**: Whether post-commit operations for BRM are enabled. The default is **false**. • **updateRequestServerTimeout**: The server timeout for update requests, in milliseconds. The default is **20000**. • **failBalanceUpdatesOnCreditCeilingBreach**: Whether balance updates fail when a credit-ceiling breach occurs. The default is **false**. • **asoCleanupOperationList**: The list of operations for ASO cleanup. The default is **TERMINATE,CANCEL**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **kafkaEnabledForNotifications**: Whether Kafka is used to publish notifications. The default is **true**. • **terminatedSessionHistoryCacheEnabled**: Whether caching of terminated-session history is enabled. The default is **false**. • **supportOriginalBeat**: Whether the original beat or timing value is preserved and supported. The default is **false**. • **walletReadRetryInterval**: The interval between wallet-read retry attempts, in milliseconds. The default is **100**. • **walletReadRetryCount**: The maximum number of wallet-read retry attempts. The default is **10**. • **useOfferPriorityDuringAlterationAgreementEvaluation**: Whether offer priority is considered when alteration agreements are evaluated. The default is **false**. • **debitSessionCheckDisabled**: Whether debit-session checking is disabled. The default is **false**. • **randomizationIntervalForUsageValidity**: The randomization interval applied to usage validity, in seconds. A value of 0 disables randomization. The default is **0**. • **populateCurrentLoanAmountsOnRef**: Whether current loan amounts are populated during reference operations. The default is **true**. • **saveNonCounterGrantDuringOngoingSession**: Whether non-counter grants are saved during an ongoing session. The default is **false**. • **checkReservationOverImpact**: Whether the system checks for reservation-over-impact conditions. The default is **false**. • **cycleForwardRenewTimeInSecond**: The cycle-forward renewal time, in seconds. A value of 0 means that no renewal time is configured. The default is **0**. • **excludedEventsForDebitRefundSessions**: The comma-separated list of event types excluded from debit refund sessions. An empty value means that no events are excluded. The default is **an empty value**. • **cleanupTransactionLockEnabled**: Whether transaction-lock cleanup is enabled. The default is **false**. • **transactionLockCleanupBatchSize**: The maximum number of transaction locks processed in each cleanup batch. The default is **200**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **transactionLockCleanupInterval**: The interval between transaction-lock cleanup operations, in milliseconds. The default is **300000**. • **enableFuiForInitiate**: Whether First Usage Initialization (FUI) is enabled for initiate operations. The default is **false**. • **replicateFirstUsageValidityInitEvent**: Whether the first-usage-validity initialization event is replicated. The default is **false**. • **aggregateCurrencyBalanceForDebitUnit**: Whether currency balances are aggregated for debit-unit operations. The default is **true**. • **aggregateNonCurrencyBalanceForDebitUnit**: Whether non-currency balances are aggregated for debit-unit operations. The default is **false**. • **enableSySessionCleanUp**: Whether session cleanup is enabled for SY service logic. The default is **false**. • **checkUsedUnitsAfterFUI**: Whether used units are checked after First Usage Initialization. The default is **false**. • **enableRatedEventsAggregationForMsccRequest**: Whether rated events are aggregated for an MSCC request. The default is **false**. • **suppressMultipleThresholdBreachNotification**: Whether multiple threshold-breach notifications are suppressed so that a single aggregated notification can be generated. The default is **false**. • **disableFederationInterceptor**: Whether the federation interceptor is disabled. The default is **false**. • **enableFederationDeadLetterQueue**: Whether the dead-letter queue for federation processing is enabled. The default is **false**. • **subscriptionIdOnStrEnabled**: Whether subscription identifiers are included in STR messages when the corresponding PCF or PCRF feature is enabled. The default is **false**. • **grantRenewalDuringAuthorization**: Whether grant renewal is performed during authorization. The default is **false**. • **rolloverConsumptionRule**: The rule used to consume rollover balances or resources. The default is **NONE**. • **sharingRetryIntervalInMillis**: The interval between sharing-operation retry attempts, in milliseconds. The default is **5**. • **reverseRateAllowedList**: The setting specifies which operations should be reverse rated. reverseRateAllowedList should have

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		INITIATE,UPDATE. The default is **INITIATE,UPDATE,DEBIT_UNIT,DEBIT_AMOUNT**. • **skipTerminateForFUA**: Whether terminate processing is skipped for Final-Unit Action (FUA). The default is **false**. • **loadLogins**: Whether login information is loaded during processing. The default is **false**. • **retryDuringLockContention**: The maximum number of retry attempts made when lock contention occurs. The default is **10**. • **enableSnrForProductCancellation**: Whether SNR processing is enabled for product cancellation. The default is **true**. • **productsConsideredForTariffSplit**: The set of products considered when applying a tariff split. The default is **ALL**. • **serviceMonitorLevel**: The diagnostic level used by the service monitor when an issue is detected. The default is **THREAD_DUMP**. • **serviceMonitorNormalCheckInterval**: The interval between service-monitor checks during normal operation, in milliseconds. The default is **5000**. • **serviceMonitorAlertCheckPeriod**: The period during which the service monitor performs alert-state checks, in milliseconds. The default is **3000**. • **serviceMonitorAlertCheckInterval**: The interval between service-monitor checks while in an alert state, in milliseconds. The default is **500**. • **serviceMonitorRestartWaitPeriod**: The amount of time the service monitor waits before restarting a service, in milliseconds. The default is **180000**. • **splitLongCyclePeriod**: Whether long billing-cycle periods are split into smaller periods. The default is **false**. • **itemAssignmentInRerating**: Whether item assignment is performed during rerating. The default is **false**. • **syntheticUpdateInInterceptor**: Whether the interceptor generates synthetic update operations. The default is **false**. • **largeGroupMode**: The processing mode used for large groups. The default is **OFF**. • **considerUninitialisedBalanceAsZeroForPolicy**: Whether an uninitialized balance is treated as zero during policy evaluation. The default is **false**. • **thresholdBasedTierSelectionMode**: The mode used to select pricing tiers when a

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		threshold is crossed. The default is **SPLIT_PERIOD_QTY**. • **policyCounterNotificationAggregationEnabled**: Whether policy counter notification aggregation is enabled. The default is **true**. • **sharingAgreementOwnerResolutionEnabled**: Whether sharing agreement owner resolution is enabled. The default is **false**. • **weblogic.jmsmodule**: The JMS module name. The default is **ECE**. • **weblogic.subdeployment**: The subdeployment name for the queue. The default is **ECEQueue**.
customerGroupConfigurations.*	**charging**	The configurations for customer groups. • **name**: The name identifies the configuration entry. Each dash (-) defines a separate customer group configuration Name of the customer group. • **clusterPreference.priority**: The priority level for the first preferred cluster. Values vary by list entry. Values in values.yaml include **1, 2**. • **clusterPreference.routingGatewayList**: The list of routing gateways associated with the cluster (likely empty if not used). The default is an empty value. • **clusterPreference.name**: The name of the first preferred cluster. Values vary by list entry. Values in values.yaml include **BRM, BRM2**.
siteConfigurations.*	**charging**	The configuration for monitoring sites. • **name**: The name of the monitoring site. Values vary by list entry. Values in values.yaml include **BRM, BRM2**. • **monitorAgentJmxSslEnabled**: Whether to enable SSL when connecting to all monitoring-agent JMX servers in this site. The default is **false**. • **monitorAgentJmxConfigurations.name**: The name of the first monitoring agent. Values vary by list entry. Values in values.yaml include **monitoringagent1, monitoringagent2**. • **monitorAgentJmxConfigurations.host**: The hostname or IP address of the monitoring agent. • **monitorAgentJmxConfigurations.jmxPort**: The JMX port used by the monitoring agent. • **monitorAgentJmxConfigurations.disableMonitor**: The flag to disable monitoring for this agent.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
ratedEventPublishers.*	**charging**	The configuration for rated event publishers. • **clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **noSQLConnectionName**: The this property defines the name of the NoSQL connection used by the publisher. The default is **noSQLConnection**. • **threadPoolSize**: The this property sets the thread pool size for the publisher. The default is **4**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
ratedEventKafkaPublishers.*	**charging**	The configuration for rated event kafka publishers. • **clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **kafkaConfigurationName**: The this property defines the name of a corresponding KafkaConfiguration. The default is **BRM**. • **topicName**: The this property sets the topicName where the rated events will be published. The default is **ece-rated-events**. • **threadPoolSize**: The number of publisher threads (string or int). The default is **4**. • **publishTimeoutSeconds**: The per-record publish timeout (seconds). The default is **5**. • **partitioningKeyStrategy**: The this property defines the partitioning key used by the publisher. Options: SESSION_ID, CUSTOMER_ID, EXTERNAL_REFERENCE. The default is **SESSION_ID**. • **reconnectBackoffIntervalMs**: The this property sets reconnect back off interval for the kafka publisher. The default is **5000**. • **reconnectBackoffMaxIntervalMs**: The this property sets the max reconnect wait time for the kafka publisher. The default is **10000**. • **extraKafkaProperties.reconnect.backoff.ms** : The ms setting specifies the configured value. The default is **175**. • **extraKafkaProperties.reconnect.backoff.max.ms**: The ms setting specifies the configured value. The default is **1000**. • **dlqTopicName**: The dead Letter Queue (DLQ) configuration (always enabled) DLQ routes terminal per-record failures to the configured topic using a non-transactional producer Required. The default is **ece-rated-events-dlq**. • **dlqPartitioningKeyStrategy**: The DLQ partition key strategy. Allowed: SESSION_ID (default), CUSTOMER_ID, EXTERNAL_REFERENCE - Falls back to SESSION_ID when chosen field is blank; value is case-sensitive. The default is **SESSION_ID**. • **dlqMaxPayloadBytes**: The max length for reason/payload in DLQ envelope; 0 means no truncation; must be >= 0. The default is **2048**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
itemAssignmentConfig.*	**charging**	The details for itemAssignmentConfig: • **itemAssignmentEnabled**: Whether item assignment is enabled. The default is **true**. • **delayToleranceIntervalInDays**: The delay tolerance interval in days setting specifies the configured value. The default is **0**. • **poidPersistenceSafeCount**: The maximum count of POIDs (point of identification) for persistence safety. The default is **12000**. • **poidIdConfigurations.schemaName**: The schema name setting specifies the configured value. The default is **1**. • **poidIdConfigurations.poidQuantity**: The maximum quantity of POIDs allowed for schema 1. The default is **2000000**.
simpleFileBasedRatedEventPublisher.*	**charging**	The configuration for simple file-based rated event publisher. • **targetFileDirectory**: The directory where target files are stored. The default is **../**.
errorHandler.*	**charging**	The details for errorHandler: • **maxNumberSystemExceptionsWithDetailedLoggingPerSecond**: The maximum number of system exceptions with detailed logging allowed per second. The default is **10**. • **maxNumberSystemExceptionsWithDetailedLoggingPerMinute**: The maximum number of system exceptions with detailed logging allowed per minute. The default is **100**. • **maxNumberSystemExceptionsWithDetailedLoggingPerHour**: The maximum number of system exceptions with detailed logging allowed per hour. The default is **300**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
notification.*	**charging**	The details for notification: • **creditCeilingBreachNotificationMode**: The notification modes for various events "NONE" mode indicates no notification will be sent for the corresponding event "ASYNCHRONOUS" mode implies that notifications will be processed independently and asynchronously. Asynchronous notification processing implies that the notification system does not block or wait for the notification process to complete before continuing with other tasks. The default is **NONE**. • **creditFloorBreachNotificationMode**: The credit floor breach notification mode setting specifies the configured value. The default is **NONE**. • **thresholdBreachNotificationMode**: The threshold breach notification mode setting specifies the configured value. The default is **NONE**. • **topUpNotificationMode**: The top up notification mode setting specifies the configured value. The default is **NONE**. • **rarNotificationMode**: The rar notification mode setting specifies the configured value. The default is **NONE**. • **externalTopUpNotificationMode**: The external top up notification mode setting specifies the configured value. The default is **NONE**. • **billingNotificationMode**: The billing notification mode setting specifies the configured value. The default is **NONE**. • **adviceOfChargeNotificationMode**: The advice of charge notification mode setting specifies the configured value. The default is **NONE**. • **replenishPoidIdNotificationMode**: The replenish POID ID notification mode setting specifies the configured value. The default is **ASYNCHRONOUS**. • **lifeCycleTransitionNotificationMode**: The life cycle transition notification mode setting specifies the configured value. The default is **NONE**. • **firstUsageValidityInitNotificationMode**: The first usage validity init notification mode setting specifies the configured value. The default is **NONE**. • **offeringUsageValidityInitNotificationMode**: The offering usage validity init notification mode setting specifies the configured value. The default is **ASYNCHRONOUS**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **spendingLimitNotificationMode**: The spending limit notification mode setting specifies the configured value. The default is **NONE**. • **aggregatedSpendingLimitNotificationMode** : The aggregated spending limit notification mode setting specifies the configured value. The default is **NONE**. • **subscriberPreferenceUpdateNotificationMode**: The subscriber preference update notification mode setting specifies the configured value. The default is **NONE**. • **rerateJobCreateNotificationMode**: The rerate job create notification mode setting specifies the configured value. The default is **ASYNCHRONOUS**. • **customEventNotificationMode**: The custom event notification mode setting specifies the configured value. The default is **NONE**. • **enrichBalanceQueryResponseMode**: The enrich balance query response mode setting specifies the configured value. The default is **NONE**. • **eventListForInternalExternalPublish**: The event list for internal external publish setting specifies the configured value. The default is **OFFERING_VALIDITY_INITIALIZATION_EVENT,EXTERNAL_TOP_UP_NOTIFICATION_EVENT**. • **kafkaPartitionForInternalExternal**: The Kafka partition for internal external setting specifies the configured value. The default is **0**. • **loanGrantNotificationMode**: The loan grant notification mode setting specifies the configured value. The default is **NONE**. • **automaticTopUpTriggerNotificationMode**: The automatic top up trigger notification mode setting specifies the configured value. The default is **NONE**. • **subscriptionCycleForwardMode**: The subscription cycle forward mode setting specifies the configured value. The default is **NONE**. • **customBrmOpCodeEventNotificationMode**: The custom BRM op code event notification mode setting specifies the configured value. The default is **NONE**. • **abortSessionRequestNotificationMode**: The abort session request notification mode setting specifies the configured value. The default is **ASYNCHRONOUS**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **userDisconnectNotificationMode**: The user disconnect notification mode setting specifies the configured value. The default is **NONE**. • **renewalGrantActivationNotificationMode**: The renewal grant activation notification mode setting specifies the configured value. The default is **NONE**. • **eventListForECEExternalTopic**: The list of events for ECE external topic. The default is an empty value. • **notificationEnrichmentConfigList.enrichName**: The enrich name setting specifies the configured value. The default is **subscriberPreferences**. • **notificationEnrichmentConfigList.enrichValue**: The enrich value setting specifies the configured value. The default is an empty value.
diameterGatewayPeerConfigurations.*	**charging**	The details for diameterGatewayPeerConfigurations: • **peerName**: The name of the Diameter gateway peer. The default is **peer1**. • **diameterGatewayAlternatePeerConfigurations.alternatePeerName**: The alternate peer name setting specifies the configured value. Values vary by list entry. Values in values.yaml include **alternatePeer1, alternatePeer2**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
expirationConfiguration.*	**charging**	The expiration configuration for various types of records. • **expiredAuditRetentionIntervalInDays**: The set the retention interval for expired audit records to 45 days. The default is **45**. • **expiredPurchasedChargesRetentionIntervalInDays**: The set the retention interval for expired purchased charges to 20 days. The default is **20**. • **expiredPurchasedAlterationRetentionIntervalInDays**: The set the retention interval for expired purchased alterations to 30 days. The default is **30**. • **expiredRatingProfileRetentionIntervalInDays**: The set the retention interval for expired rating profiles to 40 days. The default is **40**. • **defaultExpirationRetentionIntervalInDays**: The set the default retention interval for expired items to 50 days. The default is **50**. • **defaultExpiredBalanceItemRetentionIntervalInDays**: The set the default retention interval for expired balance items to 60 days. The default is **60**. • **defaultExpiredRBundleHistoryRetentionIntervalInDays**: The set the default retention interval for expired RBundle history to 30 days. The default is **30**. • **expiredAsoRetentionIntervalInDays**: The set the retention interval in days for expired ASO (Active Session Object) records to -1, indicating indefinite retention. The default is **-1**. • **expiredAsoRetentionIntervalInMinutes**: The set the retention interval in minutes for expired ASO (Active Session Object) records to -1, indicating indefinite retention. The default is **-1**. • **debitRefundSessionCleanUpIntervalInDays** : The debit refund session cleanup interval (-1 for indefinite retention). The default is **-1**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
extensions.*	**charging**	The details for extensions: • **preRatingExtension**: The extension executed before rating. The default is an empty value. • **ratingExtension**: The extension executed during rating. The default is an empty value. • **postRatingExtension**: The extension executed after rating. The default is an empty value. • **postChargingExtension**: The extension executed after charging. The default is an empty value. • **diameterGyExtension**: The extension executed for Diameter GY (Gy Interface) interactions. The default is an empty value. • **radiusAuthExtension**: The extension executed for RADIUS authentication. The default is an empty value. • **radiusAccountingExtension**: The extension executed for RADIUS accounting. The default is an empty value. • **postUpdateExtension**: The extension executed after an update operation. The default is an empty value. • **ocsBypassExtension**: The ocs bypass extension setting specifies the configured value. The default is an empty value. • **httpExtension**: The extension executed for HTTP interactions. The default is an empty value. • **brmGwExtension**: The extension executed for BRM (Billing and Revenue Management) Gateway. The default is an empty value. • **preRatingMidSessionExtension**: The extension executed before mid-session rating. The default is an empty value. • **diameterSyExtension**: The extension executed for Diameter SY (Service Logic) interactions. The default is an empty value. • **postRatingMidSessionExtension**: The extension executed after mid-session rating. The default is an empty value. • **httpSyExtension**: The HTTP Sy extension setting specifies the configured value. The default is an empty value.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
brmCdrPlugins.*	**charging**	The details for brmCdrPlugins: • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.name**: The unique name for the BRM CDR Plugin. The default is **brmCdrPlugin1**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.tempDirectoryPath**: The temporary directory path for processing. The default is **/tmp/tmp**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.doneDirectoryPath**: The directory path where processed files are moved. The default is **/home/charging/rem_input/**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.invalidRatedEventDirectoryPath**: The directory path for storing invalid rated events. The default is **/tmp/invalid**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.doneFileExtension**: The extension for files after processing is completed. The default is **.out**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.headerFileExtension**: The extension for header files. The default is **.out**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.dataFileExtension**: The extension for data files. The default is **.blk**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.ctlFileExtension**: The extension for control files. The default is **.ctl**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.headerFileDirectoryPath**: The directory path for header files. The default is **/home/charging/rem_input/**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.dataFileDirectoryPath**: The directory path for data files. The default is **/home/charging/rem_input/**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.enableInvalidRatedEvents**: Whether to enable processing of invalid rated events. The default is **false**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.sqlLoaderCharacterSet**: The character set for SQL Loader. The default is **UTF8**. • **brmCdrPluginConfigurationList.brmCdrPluginConfiguration.prepaidPartitionSet**: The partition set for prepaid CDRs. The default is **0**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
cdrFormatterPlugins.*	**charging**	The details for cdrFormatterPlugins: • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.name**: The unique name for the CDR Formatter Plugin. The default is **cdrFormatterPlugin1**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.tempDirectoryPath**: The temporary directory path for processing. The default is **/tmp/tmp**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.doneDirectoryPath**: The directory path where processed files are moved. The default is **/home/charging/cdr_input**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.doneFileExtension**: The extension for files after processing is completed. The default is **.out**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.enableKafkaIntegration**: Whether to enable Kafka integration. The default is **false**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.enableDiskPersistence**: Whether to enable disk persistence. The default is **true**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.maxCdrCount**: The maximum count of CDRs processed per batch. The default is **20000**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.staleSessionCauseForRecordClosingString**: The string indicating the cause for closing stale sessions. The default is **PARTIAL_RECORD**. • **cdrFormatterPluginConfigurationList.cdrFormatterPluginConfiguration.enableStaleSessionCleanupCustomField**: Whether to enable cleanup of stale session custom fields. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
nfProfileConfigurations.*	**charging**	The details for nfProfileConfigurations: • **instanceIdMapping**: The mapping of instance IDs. The default is an empty value. • **nfProfileConfigurationList.name**: The configuration for NF Profile "httpGateway1". The default is **httpGateway1**. • **nfProfileConfigurationList.nrfApiVersionInUri**: The JSON request payload for the NF Profile API version used in the NRF URI. The default is **v1**. • **nfProfileConfigurationList.clusterName**: The name of the cluster associated with the NF Profile. The default is **BRM**. • **nfProfileConfigurationList.nrfRestEndPointUrl**: The URL for NRF (Network Repository Function) REST endpoint. The default is an empty value. • **nfProfileConfigurationList.nrfSecondarySiteRestEndPointUrls**: The uRLs for secondary sites' NRF REST endpoints. The default is an empty value. • **nfProfileConfigurationList.nrfHttp2Enable**: Whether HTTP/2 is enabled. The default is **true**. • **nfProfileConfigurationList.nfType**: The type of NF (e.g., CHF, etc.). The default is **CHF**. • **nfProfileConfigurationList.fqdn**: The fully qualified domain name. The default is an empty value. • **nfProfileConfigurationList.ipv4Addresses**: The iPv4 addresses. The default is an empty value. • **nfProfileConfigurationList.capacity**: The capacity of the NF. The default is **0**. • **nfProfileConfigurationList.load**: The load on the NF. The default is **0**. • **nfProfileConfigurationList.supiRangeListStart**: The start of range for SUPI (Subscription Permanent Identifier). The default is **10000**. • **nfProfileConfigurationList.supiRangeListEnd**: The end of range for SUPI. The default is **10008**. • **nfProfileConfigurationList.supiRangeListPattern**: The pattern for SUPI range list. The default is **^nai-450081.+@.+org$**. • **nfProfileConfigurationList.gpsiRangeListStart**: The start of range for GPSI (Global Subscription Permanent Identifier). The default is **10000**. • **nfProfileConfigurationList.gpsiRangeListEnd**: The end of range for GPSI. The default is **10008**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **nfProfileConfigurationList.gpsiRangeListPattern**: The pattern for GPSI range list. The default is **^extid-.+@oracle1.com$**. • **nfProfileConfigurationList.plmnRangeListStart**: The start of range for PLMN (Public Land Mobile Network). The default is **100000**. • **nfProfileConfigurationList.plmnRangeListEnd**: The end of range for PLMN. The default is **333333**. • **nfProfileConfigurationList.plmnRangeListPattern**: The pattern for PLMN range list. The default is an empty value. • **nfProfileConfigurationList.nfStatus**: The status of the NF. The default is **REGISTERED**. • **nfProfileConfigurationList.heartBeatTimer**: The heartbeat timer. The default is an empty value. • **nfProfileConfigurationList.plmnListMcc**: The MCC (Mobile Country Code) list for PLMN. The default is an empty value. • **nfProfileConfigurationList.plmnListMnc**: The MNC (Mobile Network Code) list for PLMN. The default is an empty value. • **nfProfileConfigurationList.snssaisSdl**: The SD (Serving Data) list for SNSSAI (Service Network Slice Selection Assistance Information). The default is an empty value. • **nfProfileConfigurationList.snssaisSst**: The SST (Slice/Service Type) list for SNSSAI. The default is an empty value. • **nfProfileConfigurationList.perPlmnSnssaiListPlmnIdMcc**: The MCC list for per PLMN SNSSAI. The default is an empty value. • **nfProfileConfigurationList.perPlmnSnssaiListPlmnIdMnc**: The MNC list for per PLMN SNSSAI. The default is an empty value. • **nfProfileConfigurationList.perPlmnSnssaiListSst**: The SST list for per PLMN SNSSAI. The default is an empty value. • **nfProfileConfigurationList.perPlmnSnssaiListSd**: The SD list for per PLMN SNSSAI. The default is an empty value. • **nfProfileConfigurationList.nsiList**: The NSI (Network Slice Instance) list. The default is an empty value. • **nfProfileConfigurationList.interPlmnFqdn**: The FQDN for inter-PLMN communication. The default is an empty value. • **nfProfileConfigurationList.ipv6Addresses**: The iPv6 addresses. The default is an empty value. • **nfProfileConfigurationList.allowedPlmnsMcc**: The allowed MCCs (Mobile Country Codes). The default is an empty value.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **nfProfileConfigurationList.allowedPlmnsMnc**: The allowed MNCs (Mobile Network Codes). The default is an empty value. • **nfProfileConfigurationList.allowedNfTypes**: The allowed NF types. The default is an empty value. • **nfProfileConfigurationList.allowedNfDomains**: The allowed NF domains. The default is an empty value. • **nfProfileConfigurationList.allowedNssaisSst**: The allowed SSTs (Slice/Service Types). The default is an empty value. • **nfProfileConfigurationList.allowedNssaisSd**: The allowed SDs (Serving Data). The default is an empty value. • **nfProfileConfigurationList.locality**: The locality information. The default is an empty value. • **nfProfileConfigurationList.customInfo**: The custom information. The default is an empty value. • **nfProfileConfigurationList.recoveryTime**: The recovery time. The default is an empty value. • **nfProfileConfigurationList.nfServicePersistence**: The flag indicating NF service persistence. The default is an empty value. • **nfProfileConfigurationList.nfProfileChangesSupportInd**: The flag indicating NF profile changes support. The default is an empty value. • **nfProfileConfigurationList.nfProfileChangesInd**: The flag indicating NF profile changes. The default is an empty value. • **nfProfileConfigurationList.defaultNotificationSubscriptionsNotificationType**: The notification type for default notification subscriptions. The default is an empty value. • **nfProfileConfigurationList.defaultNotificationSubscriptionsCallbackUri**: The callback URI for default notification subscriptions. The default is an empty value. • **nfProfileConfigurationList.defaultNotificationSubscriptionsN1MessageClass**: The N1 message class for default notification subscriptions. The default is an empty value. • **nfProfileConfigurationList.defaultNotificationSubscriptionsN2InformationClass**: The N2 information class for default notification subscriptions. The default is an empty value. • **nfProfileConfigurationList.httpGatewayName**: The name of the associated HTTP gateway. The default is **httpgateway1**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **nfProfileConfigurationList.primaryChfInstance**: The primary NF instance (UUID). The default is an empty value. • **nfProfileConfigurationList.secondaryChfInstance**: The secondary NF instance (UUID). The default is an empty value. • **nfProfileConfigurationList.nfSetId**: The nFSet ID for grouping CHF instances for resiliency. The default is an empty value.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
nfServiceConfigurations.*	**charging**	The details for nfServiceConfigurations: • **nfServiceConfigurationList.name**: The configuration for NF Service "chf1". The default is **chf1**. • **nfServiceConfigurationList.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **nfServiceConfigurationList.serviceInstanceId**: The unique ID for the service instance. The default is **chf1**. • **nfServiceConfigurationList.serviceName**: The name of the service. The default is **nchf-convergedcharging**. • **nfServiceConfigurationList.apiVersionInUri**: The API version in URI. The default is **v1**. • **nfServiceConfigurationList.apiFullVersion**: The full API version. The default is **1.R15.1.0**. • **nfServiceConfigurationList.expiry**: The expiry timestamp. The default is **2020-12-01T18:55:08.871Z**. • **nfServiceConfigurationList.scheme**: The communication scheme (e.g., HTTP, HTTPS). The default is **http**. • **nfServiceConfigurationList.nfServiceStatus**: The status of the NF service. The default is **REGISTERED**. • **nfServiceConfigurationList.ipv4Address**: The iPv4 address of the NF service. The default is an empty value. • **nfServiceConfigurationList.transport**: The transport protocol (e.g., TCP, UDP). The default is **TCP**. • **nfServiceConfigurationList.port**: The port for communication. The default is an empty value. • **nfServiceConfigurationList.capacity**: The capacity of the NF service. The default is **50**. • **nfServiceConfigurationList.load**: The load on the NF service. The default is **5**. • **nfServiceConfigurationList.priority**: The priority of the NF service. The default is **1**. • **nfServiceConfigurationList.interPlmnFqdn**: The FQDN for inter-PLMN communication. The default is an empty value. • **nfServiceConfigurationList.ipv6Address**: The iPv6 address of the NF service. The default is an empty value. • **nfServiceConfigurationList.defaultNotificationSubscriptionsNotificationType**: The notification type for default notification subscriptions. The default is an empty value. • **nfServiceConfigurationList.defaultNotificationSubscriptionsCallbackUri**: The callback

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		URI for default notification subscriptions. The default is an empty value. • **nfServiceConfigurationList.defaultNotificationSubscriptionsN1MessageClass**: The N1 message class for default notification subscriptions. The default is an empty value. • **nfServiceConfigurationList.defaultNotificationSubscriptionsN2InformationClass**: The N2 information class for default notification subscriptions. The default is an empty value. • **nfServiceConfigurationList.allowedPlmnsMcc**: The allowed MCCs (Mobile Country Codes). The default is an empty value. • **nfServiceConfigurationList.allowedPlmnsMnc**: The allowed MNCs (Mobile Network Codes). The default is an empty value. • **nfServiceConfigurationList.allowedNfTypes** : The allowed NF types. The default is an empty value. • **nfServiceConfigurationList.allowedNfDomains**: The allowed NF domains. The default is an empty value. • **nfServiceConfigurationList.allowedNssaisSst**: The allowed SSTs (Slice/Service Types). The default is an empty value. • **nfServiceConfigurationList.allowedNssaisSd**: The allowed SDs (Serving Data). The default is an empty value. • **nfServiceConfigurationList.recoveryTime**: The recovery time. The default is an empty value. • **nfServiceConfigurationList.supportedFeatures**: The supported features. The default is an empty value. • **nfServiceConfigurationList.primaryChfServiceInstance**: The primary CHF service instance. The default is an empty value. • **nfServiceConfigurationList.secondaryChfServiceInstance**: The secondary CHF service instance. The default is an empty value. • **nfServiceConfigurationList.fqdn**: The fully qualified domain name. The default is an empty value. • **nfServiceConfigurationList.apiPrefix**: The API Prefix. The default is an empty value. • **nfServiceConfigurationList.httpGatewayName**: The name of the associated HTTP gateway. The default is **httpgateway1**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
cachePersistenceConfigurations.*	**charging**	The details for cachePersistenceConfigurations: • **cachePersistenceConfigurationList.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **cachePersistenceConfigurationList.persistenceStoreType**: The type of persistence store. The default is **OracleDB**. • **cachePersistenceConfigurationList.persistenceConnectionName**: The name of the persistence connection. The default is **oraclePersistence1**. • **cachePersistenceConfigurationList.reloadThreadPoolSize**: The size of the thread pool for reloading. The default is **10**. • **cachePersistenceConfigurationList.configLoadFromPersistence**: Whether to load configuration from persistence. The default is **false**. • **cachePersistenceConfigurationList.pricingLoadFromPersistence**: Whether to load pricing from persistence. The default is **false**. • **cachePersistenceConfigurationList.customerLoadFromPersistence**: Whether to load customer data from persistence. The default is **false**. • **cachePersistenceConfigurationList.partitionLossRecoverFromPersistence**: Whether to recover partition loss from persistence. The default is **true**. • **cachePersistenceConfigurationList.writeBehindThreadPoolSize**: The size of the thread pool for write behind. The default is **1**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
connectionConfigurations.*	**charging**	The details for connectionConfigurations: • **OraclePersistenceConnectionConfigurations.clusterName**: The clusterName determines site where ref pod is running. The default is **BRM**. • **OraclePersistenceConnectionConfigurations.schemaNumber**: The schema number. The default is **1**. • **OraclePersistenceConnectionConfigurations.name**: The name of the connection. The default is **oraclePersistence1**. • **OraclePersistenceConnectionConfigurations.dbSysDBAUser**: The username for DB system DBA. The default is **sys**. • **OraclePersistenceConnectionConfigurations.dbSysDBARole**: The role for DB system DBA. The default is **sysdba**. • **OraclePersistenceConnectionConfigurations.userName**: The username for the connection. The default is **ece**. • **OraclePersistenceConnectionConfigurations.hostName**: The hostname of the database. The default is an empty value. • **OraclePersistenceConnectionConfigurations.port**: The port number of the database. The default is **1521**. • **OraclePersistenceConnectionConfigurations.sid**: The SID (System Identifier) of the database. The default is an empty value. • **OraclePersistenceConnectionConfigurations.service**: The service name of the database. The default is an empty value. • **OraclePersistenceConnectionConfigurations.tablespace**: The tablespace for ECE. The default is **ecetable**. • **OraclePersistenceConnectionConfigurations.temptablespace**: The temporary tablespace for ECE. The default is **ecetemp**. • **OraclePersistenceConnectionConfigurations.cdrstoretablespace**: The tablespace for CDR (Call Detail Record) store. The default is **ececdrtablespace**. • **OraclePersistenceConnectionConfigurations.cdrstoreindexspace**: The tablespace for CDR index. The default is **ececdrindexspace**. • **OraclePersistenceConnectionConfigurations.jdbcUrl**: The JDBC URL for the connection. The default is an empty value. • **OraclePersistenceConnectionConfigurations.retryCount**: The number of retry attempts. The default is **3**. • **OraclePersistenceConnectionConfigurations.retryInterval**: The interval between retry attempts. The default is **1**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **OraclePersistenceConnectionConfigurations.maxStmtCacheSize**: The maximum size of the statement cache. The default is **100**. • **OraclePersistenceConnectionConfigurations.connectionWaitTimeout**: The timeout for connection wait. The default is **300**. • **OraclePersistenceConnectionConfigurations.timeoutConnectionCheckInterval**: The interval for connection check timeout. The default is **300**. • **OraclePersistenceConnectionConfigurations.inactiveConnectionTimeout**: The timeout for inactive connections. The default is **300**. • **OraclePersistenceConnectionConfigurations.databaseConnectionTimeout**: The timeout for database connections. The default is **600**. • **OraclePersistenceConnectionConfigurations.persistenceInitialPoolSize**: The initial size of the persistence pool. The default is **4**. • **OraclePersistenceConnectionConfigurations.persistenceMinPoolSize**: The minimum size of the persistence pool. The default is **4**. • **OraclePersistenceConnectionConfigurations.persistenceMaxPoolSize**: The maximum size of the persistence pool. The default is **12**. • **OraclePersistenceConnectionConfigurations.reloadInitialPoolSize**: The initial size of the reload pool. The default is **0**. • **OraclePersistenceConnectionConfigurations.reloadMinPoolSize**: The minimum size of the reload pool. The default is **0**. • **OraclePersistenceConnectionConfigurations.reloadMaxPoolSize**: The maximum size of the reload pool. The default is **20**. • **OraclePersistenceConnectionConfigurations.ratedEventInitialPoolSize**: The rated event initial pool size setting specifies the configured value. The default is **0**. • **OraclePersistenceConnectionConfigurations.ratedEventMinPoolSize**: The rated event min pool size setting specifies the configured value. The default is **1**. • **OraclePersistenceConnectionConfigurations.ratedEventMaxPoolSize**: The rated event max pool size setting specifies the configured value. The default is **1**. • **OraclePersistenceConnectionConfigurations.ratedEventFormatterInitialPoolSize**: The initial size of the rated event formatter pool. The default is **6**. • **OraclePersistenceConnectionConfigurations.ratedEventFormatterMinPoolSize**: The

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		minimum size of the rated event formatter pool. The default is **6**. • **OraclePersistenceConnectionConfigurations.ratedEventFormatterMaxPoolSize**: The maximum size of the rated event formatter pool. The default is **6**. • **OraclePersistenceConnectionConfigurations.ratedEventTablePartitionByMinute**: The partition time for rated event table. The default is **5**. • **OraclePersistenceConnectionConfigurations.ratedEventTableStorageInitial**: The initial storage size for rated event table. The default is **104857600**. • **OraclePersistenceConnectionConfigurations.ratedEventTableStorageNext**: The next storage size for rated event table. The default is **104857600**. • **OraclePersistenceConnectionConfigurations.ratedEventTableSubpartitions**: The number of subpartitions for rated event table. The default is **32**. • **OraclePersistenceConnectionConfigurations.dbSSLEnabled**: Whether DB SSL is enabled. The default is **false**. • **OraclePersistenceConnectionConfigurations.tlsVersion**: The TLS version used for database connection. The default is **1.3**. • **OraclePersistenceConnectionConfigurations.dbSSLType**: The type of database SSL configuration (e.g., "twoway" for mutual SSL). The default is **oneway**. • **OraclePersistenceConnectionConfigurations.sslServerCertDN**: The distinguished Name (DN) of the server SSL certificate for database connection. The default is **DC=local,DC=oracle,CN=pindb**. • **OraclePersistenceConnectionConfigurations.trustStoreLocation**: The location of the trust store. The default is **cwallet.sso**. • **OraclePersistenceConnectionConfigurations.trustStoreType**: The type of trust store. The default is **SSO**. • **OraclePersistenceConnectionConfigurations.walletLocation**: The location of the wallet. The default is **/home/charging/wallet/ecewallet/**. • **OraclePersistenceConnectionConfigurations.cdrStorePartitionCount**: The number of partitions for CDR store. The default is **32**. • **OraclePersistenceConnectionConfigurations.queryTimeout**: The query timeout. The default is **5**. • **OraclePersistenceConnectionConfigurations.oraFilesLocation**: The location of Oracle

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		files used for database connections or configurations (e.g., Oracle Database Resource Manager files). The default is **/home/charging/ext/ora_files/ece/**. • **OraclePersistenceConnectionConfigurations.extPersistenceSecret**: The name of the external secret containing Persistence passwords, if present. • **OraclePersistenceConnectionConfigurations.extPersistenceDBSSLWalletSecret**: The name of the external Kubernetes secret containing the ECE database SSL TrustStore. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information. • **NoSQLConnectionConfigurations.clusterName**: The configuration for NoSQL Connection Configurations. clusterName determines site where ref pod is running. The default is **BRM**. • **NoSQLConnectionConfigurations.name**: The name of the connection. The default is **noSQLConnection**. • **NoSQLConnectionConfigurations.dataStoreConnection**: The data store connection details. The default is an empty value. • **NoSQLConnectionConfigurations.dataStoreConnectionHost**: The data store connection host setting specifies the configured value. The default is an empty value. • **NoSQLConnectionConfigurations.dataStoreConnectionAlias**: The data store connection alias setting specifies the configured value. The default is an empty value. • **NoSQLConnectionConfigurations.dataStoreConnectionIP**: The data store connection IP setting specifies the configured value. The default is an empty value. • **NoSQLConnectionConfigurations.dataStoreName**: The name of the data store. The default is **kvstore**. • **NoSQLConnectionConfigurations.noSQLTimeout**: The timeout for NoSQL operations. The default is **5**. • **NoSQLConnectionConfigurations.retryCount**: The number of retry attempts. The default is **5**. • **NoSQLConnectionConfigurations.cdrStorePartitionCount**: The number of partitions for CDR store. The default is **32**. • **BRMConnectionConfiguration.name**: The configuration for BRM Connection. The default is **brmConnection**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **BRMConnectionConfiguration.hostName**: The hostname of the BRM. The default is **cm**. • **BRMConnectionConfiguration.protocol**: The protocol used for connection. The default is **pcp**. • **BRMConnectionConfiguration.loginName**: The login name for authentication. The default is **root.0.0.0.1**. • **BRMConnectionConfiguration.cmPort**: The port number for CM (Connection Manager). The default is **11960**. • **BRMConnectionConfiguration.loginType**: The login type. The default is **1**. • **BRMConnectionConfiguration.numberOfConnections**: The number of connections. The default is **10**. • **BRMConnectionConfiguration.sslEnabled**: The flag indicating SSL enablement. The default is **1**. • **BRMConnectionConfiguration.failOverConnectionUrls**: The failover connection URLs. The default is an empty value. • **BRMConnectionConfiguration.eceWalletLocation**: The location of ECE wallet. The default is **/home/charging/wallet/ecewallet/**. • **BRMConnectionConfiguration.brmwallet**: The primary BRM client wallet filename. The chart resolves it under /home/charging/wallet/brmwallet/client. The default is **cwallet.sso**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
brsConfigurations.*	**charging**	The details for brsConfigurations: • **brsConfigurationList.acceptablePendingCount**: The acceptable pending count. The default is **10**. • **brsConfigurationList.highPriorityBatchSize**: The batch size for high priority requests. The default is **2**. • **brsConfigurationList.highPriorityBatchTimeout**: The timeout for high priority batch. The default is **30**. • **brsConfigurationList.highPriorityThreadPoolSize**: The size of thread pool for high priority requests. The default is **-1**. • **brsConfigurationList.name**: The name of the configuration. The default is **default**. • **brsConfigurationList.overloadProtection**: Whether overload protection is enabled. The default is **false**. • **brsConfigurationList.pendingRequestHeapUsageThresholdPercentage**: The threshold percentage for pending request heap usage. The default is **30**. • **brsConfigurationList.regularPriorityBatchSize**: The batch size for regular priority requests. The default is **1**. • **brsConfigurationList.regularPriorityBatchTimeout**: The timeout for regular priority batch. The default is **50**. • **brsConfigurationList.regularPriorityThreadPoolSize**: The size of thread pool for regular priority requests. The default is **200**. • **brsConfigurationList.reportingWindowSize**: The reporting window size. The default is **30**. • **brsConfigurationList.responseTimeout**: The timeout for response. The default is **10**. • **brsConfigurationList.thresholdLatency**: The threshold latency. The default is **10**. • **brsConfigurationList.throttleProcessingPerCustomer**: Whether the throttle processing per customer option is enabled. The default is **false**. • **brsConfigurationList.httpConnectionPoolMax**: The maximum size of HTTP connection pool. The default is **500**. • **brsConfigurationList.updateResponseMaxWaitTime**: The maximum time to wait for update response. The default is **10000**. • **brsConfigurationList.routingGatewayHealthCheckTimeout**: The timeout (in milliseconds) for health checks performed on the routing gateway. The default is **2000**. • **brsConfigurationList.routingGatewayHealthCheckInterval**: The interval (in milliseconds)

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		between health check attempts for the routing gateway. The default is **20000**. • **brsConfigurationList.skipActiveActivePreferredSiteRouting**: Whether to skip preferred site routing in an active-active setup. Set to "true" to skip preferred site routing. The default is **false**. • **brsConfigurationList.ingressRequestTimeoutMs**: The additional number of milliseconds that BRS will allow before evicting/aborting requests that have not yet been sent to the ECE Server. The default is **3000**. • **brsConfigurationList.requestMaxRetry**: The maximum number of times BRS will retry processing a request that has not expired. The default is **3**. • **brsConfigurationList.requestRetryWaitTimeMs**: The additional number of milliseconds that BRS will wait between retries. The default is **5**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
messagePriorityConfigurations.*	**charging**	The details for messagePriorityConfigurations: • **messagePriorityConfiguration.name**: The name setting specifies the configured value. The default is **default**. • **messagePriorityConfiguration.priorityMap.name**: The name setting specifies the configured value. Values vary by list entry. Values in values.yaml include **subscribeNotificationRequestSubscribe, subscribeNotificationRequestUnsubscribe, policySessionRequestInitial, policySessionRequestUpdate, policySessionRequestTerminate, spendingLimitReportRequestInitial, spendingLimitReportRequestUpdate, spendingLimitReportRequestTerminate, authenticationQuery, balanceQuery, userDataRequest, usageInitiate, usageUpdate, usageTerminate, usageCancel, usageDebitUnit, usageDebitAmount, usageRefundUnit, usageRefundAmount, usagePriceEnquiry, usageStartAccounting, usageUpdateAccounting, usageEvent, usageTerminateAmount, balanceUpdate, externalTopUpUpdate, createCustomer, modifyCustomer, publicIdentifierUpdate, createProfile, modifyProfile, deleteProfile, createOfferProfile, updateOfferProfile, deleteOfferProfile, productUpdate, createProduct, sharingAgreementParticipateCreate, sharingAgreementParticipateModify, sharingAgreementParticipateDelete, alterationSharingAgreementCreate, alterationSharingAgreementModify, alterationSharingAgreementDelete, distributionSharingAgreementCreate, distributionSharingAgreementModify, distributionSharingAgreementDelete, profileSharingAgreementCreate, profileSharingAgreementModify, profileSharingAgreementDelete, billingUnitUpdate, lifecycleStateUpdate, offeringExternalIdUpdate, prepareToRerate, prepareToRerateCustLock, prepareToRerateCustLockPending, rerateCompleted, rerateCompletedCustInSyncPending, rerateCompletedCustSyncStatePrepare, rerateCompletedCustSyncState, rerateCompletedSponsorBackoutPrepare, rerateCompletedSponsorBackout, rerateCompletedCustReplay, rerateCompletedDependentCustFlush, rerateCompletedCustUnlock, rerateCustFailure, ammHoldCdr,**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		ammMigrateAcct, ammMigrateSource, ammMigrateDest, ammResumeCdr, serviceTransfer, balanceChange, deleteCustomer, ecePostCommit, autoTopUpCreate, autoTopUpDelete, outstandingAmountUpdate, createBalReservation, updateBalReservation, deleteBalReservation, listBalReservation, getBalReservation, groupSharingMonitorAgreementCreate, groupSharingMonitorAgreementDelete, groupSharingMonitorAgreementModify, productSharingAgreementCreate, productSharingAgreementModify, productSharingAgreementDelete, consumedReservationTimepoint. • **messagePriorityConfiguration.priorityMap.priority**: The priority setting specifies the configured value. Values vary by list entry. Values in values.yaml include **LOW, HIGH, MEDIUM**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
overloadConfigurations.*	**charging**	The details for overloadConfigurations: • **overloadConfigurationList.name**: The name of the Overload configuration. The default is **default**. • **overloadConfigurationList.enabled**: Whether overload configuration is enabled. The default is **false**. • **overloadConfigurationList.compositionType**: The composite Type. The default is **OR**. • **overloadConfigurationList.thresholdOverloadMonitorList.name**: The name of the Threshold Overload Monitor. The default is **BRS Monitor**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.name**: The name of the Threshold Monitors. The default is **BRS Latency**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.type**: The type of the Threshold Monitors. The default is **BRS_LATENCY**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.updateIntervalMs**: The update interval in ms. The default is **100**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.initialDelayMs**: The initial delay in ms. The default is **10000**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.defaults.entryDelayMs**: The entry delay in ms by default. The default is **1000**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.defaults.exitDelayMs**: The exit delay in ms by default. The default is **500**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.defaults.alpha**: The smoothing factor for the EWMA calculation by default. The default is **0.1**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.defaults.adaptivityFactor**: The factor used to adjust the dynamic thresholds by default. The default is **0.5**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelOne.entryThreshold**: The initial entry threshold for Level One. The default is **2000**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelOne.exitThreshold**: The initial exit threshold for Level One. The default is **1250**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelOne.entryDelayMs**: The entry delay in ms for Level One. The default is **1000**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelOne.exitDelayMs**: The exit delay in ms for Level One. The default is **500**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelOne.alpha**: The smoothing factor for the EWMA calculation for level one. The default is **0.1**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelOne.adaptivityFactor**: The factor used to adjust the dynamic thresholds for level One. The default is **0.5**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelTwo.entryThreshold**: The initial entry threshold for level two. The default is **2500**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelTwo.exitThreshold**: The initial exit threshold for level two. The default is **2250**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelTwo.entryDelayMs**: The entry delay in ms for level two. The default is **1000**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelTwo.exitDelayMs**: The exit delay in ms for level two. The default is **500**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelTwo.alpha**: The smoothing factor for the EWMA calculation for level two. The default is **0.1**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelTwo.adaptivityFactor**: The factor used to adjust the dynamic thresholds for level two. The default is **0.5**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelThree.entryThreshold**: The initial entry threshold for level three. The default is **3000**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelThree.exitThreshold**: The initial exit threshold for level three. The default is **2700**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelThree.entryDelayMs**: The entry delay in ms for Level Three. The default is **1000**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelThre**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		e.exitDelayMs: The exit delay in ms for Level Three. The default is **500**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelThree.alpha**: The smoothing factor for the EWMA calculation for Level Three. The default is **0.1**. • **overloadConfigurationList.thresholdOverloadMonitorList.thresholdMonitors.levelThree.adaptivityFactor**: The factor used to adjust the dynamic thresholds for Level Two. The default is **0.5**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
kafkaConfigurations.*	**charging**	The details for kafkaConfigurations: • **kafkaConfigurationList.name**: The name of the Kafka configuration. The default is **BRM**. • **kafkaConfigurationList.hostname**: The hostname of Kafka. The default is **ece-kafka:9093**. • **kafkaConfigurationList.topicName**: The topic name. The default is **ECENotifications**. • **kafkaConfigurationList.suspenseTopicName**: The suspense topic name. The default is **ECESuspenseQueue**. • **kafkaConfigurationList.failureTopicName**: The failure topic name. The default is **EceFailureQueue**. • **kafkaConfigurationList.overageTopicName**: The overage topic name. The default is **ECEOverageTopicName**. • **kafkaConfigurationList.partitions**: The number of partitions. The default is **200**. • **kafkaConfigurationList.failurePartitions**: The number of failure partitions. The default is **200**. • **kafkaConfigurationList.externalTopicEnabled**: Whether the external Kafka topic is enabled. The default is **false**. • **kafkaConfigurationList.externalTopicName**: The external topic name. The default is **EceExtNotifications**. • **kafkaConfigurationList.externalPartitions**: The number of external partitions. The default is **15**. • **kafkaConfigurationList.walletLocation**: The location of wallet. The default is **/home/charging/wallet/ecewallet/**. • **kafkaConfigurationList.kafkaRGWTopicName**: The Kafka Radius topic name. The default is **ECERadiusNotifications**. • **kafkaConfigurationList.kafkaProducerReconnectionInterval**: The reconnection interval for Kafka producer. The default is **2000**. • **kafkaConfigurationList.kafkaProducerReconnectionMax**: The maximum reconnection time for Kafka producer. The default is **30000**. • **kafkaConfigurationList.kafkaDGWReconnectionInterval**: The reconnection interval for Kafka DGW. The default is **2000**. • **kafkaConfigurationList.kafkaDGWReconnectionMax**: The maximum reconnection time for Kafka DGW. The default is **30000**. • **kafkaConfigurationList.kafkaBRMReconnectionInterval**: The reconnection interval for Kafka BRM. The default is **2000**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **kafkaConfigurationList.kafkaBRMReconnectionMax**: The maximum reconnection time for Kafka BRM. The default is **30000**. • **kafkaConfigurationList.kafkaHTTPReconnectionInterval**: The reconnection interval for Kafka HTTP. The default is **2000**. • **kafkaConfigurationList.kafkaHTTPReconnectionMax**: The maximum reconnection time for Kafka HTTP. The default is **30000**. • **kafkaConfigurationList.persistFailedRequestsToKafkaTopic**: Whether failed requests should be persisted to a Kafka topic (set to "true" to enable). The default is **false**. • **kafkaConfigurationList.kafkaCDRTopicName**: The topic name for Kafka CDR. The default is **ChfChargingRecords**. • **kafkaConfigurationList.kafkaCDRPartitions**: The number of partitions for Kafka CDR. The default is **1**. • **kafkaConfigurationList.kafkaSSLEnabled**: Whether SSL is enabled for Kafka CDR communication. The default is **false**. • **kafkaConfigurationList.kafkaTrustStoreLocation**: The location of trust store for Kafka CDR. The default is an empty value. • **kafkaConfigurationList.extKafkaTrustStoreSecret**: The name of the external Kubernetes secret containing the server Kafka SSL keystore. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information. • **kafkaConfigurationList.kafkaCDRCompressionType**: The compression type for Kafka CDR. The default is **gzip**. • **kafkaConfigurationList.kafkaRGWReconnectionIntervalMillis**: The reconnection interval for Kafka RGW. The default is **2000**. • **kafkaConfigurationList.kafkaRGWReconnectionMaxMillis**: The maximum reconnection time for Kafka RGW. The default is **30000**. • **kafkaConfigurationList.kafkaCustomPartitionAssignorEnabled**: Whether the custom Kafka partition assignor is enabled. The default is **false**. • **kafkaConfigurationList.kafkaFederationIssuesTopicName**: The Kafka DLQ for federation issues. The default is **federationIssues**. • **kafkaConfigurationList.kafkaFederationIssuesReconnectionInterval**: The Kafka federation issues reconnection interval setting specifies the configured value. The default is **2000**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **kafkaConfigurationList.kafkaFederationIssuesReconnectionMax**: The Kafka federation issues reconnection max setting specifies the configured value. The default is **30000**. • **kafkaConfigurationList.kafkaOAuthEnabled** : Whether Kafka OAuth 2.0 authentication is enabled between brokers and clients. The default is **false**. • **kafkaConfigurationList.kafkaOAuthClientId**: The Kafka client ID of the configured client. The default is an empty value. • **kafkaConfigurationList.kafkaOAuthTokenEndpoint**: The Kafka OAuth token endpoint URL. The default is an empty value. • **kafkaConfigurationList.kafkaOAuthClientScope**: The Kafka OAuth optional client scope. The default is an empty value. • **kafkaConfigurationList.kafkaOAuthEndpointHttpsEnabled**: Whether to enable the Kafka truststore for the OAuth token endpoint when it uses HTTPS. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
loader.*	**migration**	The details for loader: • **pricingDataDirectory**: The setting specifies the directory where pricing data files are located. The default is **/home/charging/opt/ECE/oceceserver/sample_data/pricing_data**. • **configObjectsDataDirectory**: The setting points to the directory containing configuration objects for end-to-end specifications. The default is **/home/charging/opt/ECE/oceceserver/sample_data/config_data/specifications/ece_end2end**. • **productOfferingCrossRefFilePath**: The file path for the product offering cross-reference XML, mapping product offerings between systems. The default is **/home/charging/opt/ECE/oceceserver/sample_data/crossref_data/ProductXRefSample.xml**. • **customerDataDirectory**: The directory containing customer data files. The default is **/home/charging/opt/ECE/oceceserver/sample_data/customer_data**. • **customerXmlPattern**: The pattern used to identify customer XML files within the customer data directory. The default is **customer**. • **remoteWmThreads**: The number of threads for remote work manager to process tasks concurrently. The default is **1**. • **batchSize**: The number of records processed in one batch. The default is **5000**. • **dbConnections**: The number of database connections to use. The default is **1**. • **dbFetchSize**: The number of records to fetch from the database in one go. The default is **5000**. • **payloadConfigFilePath**: The placeholder for the payload configuration file path. The default is **@PAYLOAD_CONFIG_FILE_PATH@**. • **selectiveMigrationMode**: Whether migration is running in a mode that selectively migrates data. The default is **false**. • **initialCustomerLoadFilterQuery**: The SQL queries for filtering customers during initial and incremental loads. The default is an empty value. • **incrementalCustomerLoadFilterQuery**: The incremental customer load filter query setting specifies the configured value. The default is an empty value. • **customerGroupList**: The customerGroupList - values must be separated by Commas -

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		Used only if isFederation is true. The default is an empty value.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
pricingUpdater.*	**migration**	The configurations for connecting to a server to update pricing information. • **hostName**: The hostname of the Pricing Updater service. The default is **pdc-domain-adminserver**. • **port**: The port number of the Pricing Updater service. The default is **8001**. • **userName**: The username for authentication with the Pricing Updater service. The default is **weblogic**. • **connectionURL**: The connection URL for the Pricing Updater service. The default is **t3://pdc-domain-adminserver:8001**. • **connectionRetryCount**: The retry logic for establishing the connection. The default is **1073741823**. • **connectionRetrySleepInterval**: The connection retry sleep interval setting specifies the configured value. The default is **10000**. • **connectionFactory**: The connection factory name used by the Pricing Updater service. The default is **JobDispatcher/MyXAQCF**. • **workItemQueueName**: The name of the work item queue used by the Pricing Updater service. The default is **BPA/ECE_WorkItemQueue**. • **workResultQueueName**: The name of the work result queue used by the Pricing Updater service. The default is **JobDispatcher/WORKRESULTDATA_QUEUE**. • **protocol**: The configuration for the communication protocol and context factory. The default is **t3://**. • **initialContextFactory**: The initial context factory setting specifies the configured value. The default is **weblogic.jndi.WLInitialContextFactory**. • **requestTimeOut**: The timeout for requests in milliseconds. The default is **3000**. • **keyStoreLocation**: The key store location setting specifies the configured value. The default is an empty value. • **extPDCKeyStoreSecret**: The name of the external Kubernetes secret containing the server PDC SSL keystore. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information. • **requestSpecXMLGenarationEnabled**: Whether request specification XML generation is enabled. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **requestSpecXMLGenarationLocation**: The location to store generated request specification XML files (if enabled). The default is **../config/generated-request-specifications/**.
common.*	**testtools**	The details for common: • **numCustomers**: The general settings for test execution. Total number of customers to simulate. The default is **1000**. • **startCustomerId**: The starting ID for customer generation. The default is **6500000000**. • **numTxCustomers**: The number of transactional customers. The default is **100**. • **startTxCustomerId**: The starting ID for transactional customers. The default is **4900000000**. • **numProducts**: The number of products to simulate. The default is **1**. • **clusterGroup**: The cluster group setting specifies the configured value. The default is **BRM-Dev**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
simulators.*	**testtools**	The details for simulators: • **simulatorConfig.name**: The configuration for the ECE invocation simulator. The default is **ece-invocation**. • **simulatorConfig.duration**: The duration of the simulation. The default is **30**. • **simulatorConfig.throughput**: The target operations per second. The default is **200**. • **simulatorConfig.usageIC**: The distribution of different types of usage. - IC (Information Collection): Percentage of Information Collection operations - IUT (Information Update): Percentage of Information Update operations - T (Transactions): Percentage of Transaction operations - P (Policy): Percentage of Policy operations. The default is **20**. • **simulatorConfig.usageIUT**: The usage IUT setting specifies the configured value. The default is **60**. • **simulatorConfig.usageT**: The usage T setting specifies the configured value. The default is **20**. • **simulatorConfig.usageP**: The usage P setting specifies the configured value. The default is **0**. • **simulatorConfig.usageTraffic**: The distribution of traffic types: - usageTraffic: Percentage of usage-related traffic - queryTraffic: Percentage of query-related traffic - policyTraffic: Percentage of policy-related traffic. The default is **100**. • **simulatorConfig.queryTraffic**: The query traffic setting specifies the configured value. The default is **0**. • **simulatorConfig.policyTraffic**: The policy traffic setting specifies the configured value. The default is **0**. • **simulatorConfig.queryBalance**: The balance Query Mode: How balance information is queried (SUMMARY, DETAIL) Likely a reference to an index for query mode options. The default is **0**. • **simulatorConfig.balanceQueryMode**: The balance query mode setting specifies the configured value. The default is **SUMMARY**. • **simulatorConfig.querySubscriberPreference**: The options related to subscriber preferences for queries Likely a reference to an index for preference options. The default is **0**. • **simulatorConfig.queryAuthentication**: The option to enable query authentication (true/

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		false) Likely 0 for disabled, 1 for enabled. The default is **0**. • **simulatorConfig.policySubscriptionTraffic**: The distribution of policy subscription traffic types: - policySubscriptionTraffic: Percentage of policy subscription traffic - policySpendingLimitNotificationTraffic: Percentage of spending limit notification traffic - policySessionTraffic: Percentage of policy session traffic. The default is **30**. • **simulatorConfig.policySpendingLimitNotificationTraffic**: The policy spending limit notification traffic setting specifies the configured value. The default is **50**. • **simulatorConfig.policySessionTraffic**: The policy session traffic setting specifies the configured value. The default is **20**. • **simulatorConfig.policySubscriberPreferences**: The comma-separated list of policy subscriber preferences. The default is **Channel;Language**. • **simulatorConfig.policyCounters**: The comma-separated list of policy counters to track. The default is **FREE_MIN;USAGE_COUNTER**. • **simulatorConfig.updateBalance**: The percentage of update balance operations. The default is **50**. • **simulatorConfig.updateTypeCreateCustomer**: The percentage of update type create customer operations. The default is **20**. • **simulatorConfig.updatePui**: The percentage of PUI (Personal Unblocking Key) update operations. The default is **30**. • **simulatorConfig.writeLogPercentage**: The percentage of log writes. The default is **10**. • **simulatorConfig.productType**: The type of product being simulated (e.g., TelcoGsmTelephony). The default is **TelcoGsmTelephony**. • **simulatorConfig.sessionStartDate**: The date and time to start the simulated sessions. The default is **06/02/2020 10:45:00**. • **simulatorConfig.tokenUnits**: The number of token units per session. The default is **10**. • **simulatorConfig.mscc**: Whether to enable Multi Service Control Center. The default is **false**. • **simulatorConfig.useCondensedPeriodBetweenOperationsInSession**: Whether to use condensed period between operations in a session (true/false) Likely 0 for disabled, 1 for enabled. The default is **false**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **simulatorConfig.txTraffic**: The percentage of TX (Transaction) traffic. The default is **0**. • **simulatorConfig.latencyTrackingResetDelayInSeconds**: The delay in seconds before resetting latency tracking. The default is **10**. • **sessionUnitConfig.sessionUnits**: The number of session units. The default is **15**. • **sessionUnitConfig.percentage**: The percentage of operations that belong to sessions. The default is **100**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
loader.*	**testtools**	The details for loader: • **customerBatchSize**: The loader settings for customer data preparation. Number of customers per batch. The default is **500**. • **customerThreads**: The number of threads for customer data loading. The default is **2**. • **numCreditProfiles**: The number of credit profiles to consider. The default is **4**. • **numSubscriptionProductOfferings**: The configuration for subscription and alteration offerings. Number of subscription product offerings. The default is **1**. • **purchasedOfferingsPerProduct**: The number of purchased offerings per product. The default is **1**. • **rum**: The calculation method for RUM (Rate Unit of Measure). The default is **end - start**. • **customerLoadMode**: The subscription rate graph configuration Customer load mode. Use "local" for the loader pod or "distributed" to schedule work on ECS pods. The default is **distributed**. • **customerWorkSize**: The number of customers assigned to one distributed work item. The default is **50000**. • **remoteWmThreads**: The number of WorkManager threads per ECS server member for distributed customer loading. The default is **2**. • **poidCompatibleIds**: Whether to generate numeric POID id0-compatible loader ids instead of Cust#/Pro#/Bal#/PCO_/PAO_ test ids. The default is **true**. • **subscriptionGraphs**: The subscription graphs setting specifies the configured value. The default is **#name=m4 [rate:linearRate(0.05/ mn),balance_element:"USD"]**. • **systemAlterationGraphs**: The system alteration rate graph configuration. The default is **[rate:fixedAlteration(-0.10),balance_element:"USD"]**. • **numSystemAlterationOfferings**: The number of system alteration offerings. The default is **1**. • **initialCustomerBalanceQuantity**: The initial balance quantity assigned to generated customer balances. The default is **-2000000000**. • **specFilesDirectory**: The directory containing specification files. The default is **/home/ charging/opt/ECE/oceceserver/**

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		sample_data/config_data/specifications/ece_simple.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
customerGenerator.*	**testtools**	The details for customerGenerator: • **customersPerFile**: The configuration for generating customer data files. Number of customers per generated file. The default is **500**. • **balanceAmount**: The starting balance for generated customers. The default is **-100**. • **balanceElementId**: The element ID for balance (e.g., currency code). The default is **840**. • **productOfferings**: The product offering for customers. The default is **Milestone4Sample4ChargeOffering**. • **alterationOfferings**: The alteration offering for customers. The default is **AO2_PercentDisc**. • **distributionOfferings**: The distribution offering for customers. The default is **CS50%Sharing**. • **shareAlterationAgreements**: Whether alteration agreements are shared. The default is **true**. • **productTypes**: The type of product/service offered. The default is **/service/telco/gsm/telephony**. • **destinationDir**: The directory to save generated customer data files. The default is **..** • **balanceItemsPerBalance**: The number of balance items per balance. The default is **1**. • **aliasesPerProduct**: The number of aliases per product. The default is **1**. • **productRatingProfiles**: The product rating profile configuration. The default is **PFAF**. • **customerRatingProfiles**: The customer rating profile configuration. The default is **CFAF**. • **numAuditRevisions**: The number of audit revisions. The default is **0**. • **validateXml**: Whether to validate generated XML files. The default is **true**. • **isTemplateDriven**: Whether generation is template-driven. The default is **false**. • **customerTemplateFile**: The path to customer template file. The default is **../sample_data/customer_data/customer_templates/cust_template_simple.xml**. • **txCustomerTemplateFile**: The path to customer template file for sharing transactions. The default is **../sample_data/customer_data/customer_templates/cust_template_sharing.xml**. • **multipleBalances**: Whether multiple balances are used. The default is **true**. • **extendedCustomerAttributes**: The additional attributes for customers and products.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		Additional attributes for customers. The default is **ACCOUNT_TAG: GoldCustomer**. • **extendedProductAttributes**: The additional attributes for products. The default is **SERVICE_ID: Tel0001,TELCO_INFO.MSISDN: 1231231234**.
filePath	**subscriberTrace**	The file path where the subscriber trace configuration is stored. It indicates the XML file location that contains settings for subscriber tracing. The default is **/home/charging/config/subscriber-trace.xml**.
logMaxSubscribers	**subscriberTrace**	The setting specifies the maximum number of individual subscribers that the system will track simultaneously in its logs. The default is **100**.
logMaxSubscriberSessions	**subscriberTrace**	The setting defines the maximum number of sessions per subscriber that can be logged concurrently. The default is **24**.
logExpiryWaitTime	**subscriberTrace**	The setting indicates the duration after which the logged data for a subscriber is considered old and can be purged from the system. The default is **1**.
logCleanupInterval	**subscriberTrace**	The setting specifies the frequency (in days) at which the system will perform a cleanup operation to remove expired logs based on the logExpiryWaitTime. The default is **2**.
logLevel	**subscriberTrace**	The setting sets the granularity of the logs captured for subscriber tracing. A level of “DEBUG” is the most detailed, capturing a comprehensive range of information about each subscriber session. The default is **DEBUG**.
subscriberList	**subscriberTrace**	The setting specifies a list of subscribers to be explicitly traced. The default is an empty value.
logger.*	**log4j2**	The details for logger.
rootDir	**eceproperties**	The root directory of ECE server installation. The default is **/home/charging/opt/ECE/ocecesever**.
logsDir	**eceproperties**	The directory for ECE server logs. The default is **/home/charging/opt/ECE/ocecesever/logs**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
NotificationQueue.*	**JMSConfiguration**	The details for NotificationQueue: • **Cluster**: The identifier for the BRM (Billing and Revenue Management) cluster. The default is **BRM**. • **HostName**: The hostname of the JMS server. • **Port**: The port on which JMS server is listening. • **UserName**: The username for authentication. The default is **weblogic**. • **Protocol**: The communication protocol used. The default is **t3://**. • **ConnectionURL**: The full URL for connection. • **ConnectionRetryCount**: The number of times to retry connection on failure. The default is **10**. • **KeyStoreLocation**: The location of the SSL keystore for secure connections. The default is **client.jks**. • **extJMSKeyStoreSecret**: The names of the external Kubernetes secrets for the gateway queues. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information. • **EceWalletLocation**: The location of the ECE wallet for secure transactions. The default is **/home/charging/wallet/ecewallet/**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
BRMGatewayNotificationQueue.*	**JMSConfiguration**	The configuration specifies the settings for this notification queue. • **Cluster**: The cluster setting specifies the configured value. The default is **BRM**. • **HostName**: The host name setting specifies the host for the connection. • **Port**: The port setting specifies the service port. • **UserName**: The user name setting specifies the configured value. The default is **weblogic**. • **Protocol**: The protocol setting specifies the configured value. The default is **t3://**. • **ConnectionURL**: The connection URL setting specifies the configured value. • **KeyStoreLocation**: The key store location setting specifies the configured value. The default is **client.jks**. • **extJMSKeyStoreSecret**: The names of the external Kubernetes secrets for the gateway queues. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information. • **EceWalletLocation**: The ECE wallet location setting specifies the configured value. The default is **/home/charging/wallet/ecewallet/**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
DiameterGatewayNotificationQueue.*	**JMSConfiguration**	The configuration specifies the settings for this notification queue. • **Cluster**: The cluster setting specifies the configured value. The default is **BRM**. • **HostName**: The host name setting specifies the host for the connection. • **Port**: The port setting specifies the service port. • **UserName**: The user name setting specifies the configured value. The default is **weblogic**. • **Protocol**: The protocol setting specifies the configured value. The default is **t3://**. • **ConnectionURL**: The connection URL setting specifies the configured value. • **KeyStoreLocation**: The key store location setting specifies the configured value. The default is **client.jks**. • **extJMSKeyStoreSecret**: The names of the external Kubernetes secrets for the gateway queues. See " Using External Kubernetes Secrets for Certificates and Wallets " in *BRM Cloud Native System Administrator's Guide* for more information. • **EceWalletLocation**: The ECE wallet location setting specifies the configured value. The default is **/home/charging/wallet/ecewallet/**.
jdbcConfigurations.*	**REFPluginConfiguration**	The schema-specific JDBC connection settings. Add one entry under jdbcConfigurations for each schema, all fields are compulsory. • **connectionURL**: The connection URL setting specifies the configured value. • **user**: The user setting specifies the configured value. The default is **pin**. • **wallet_entry_name**: The wallet entry name setting specifies the configured value. The default is **brm.db.pwd**.
wallet_location	**REFPluginConfiguration**	The location of the database wallet for secure connections. The default is **/home/charging/wallet/ecewallet/**.
validateSchema	**REFPluginConfiguration**	Whether to validate database schema on connection. The default is **true**.
initialPoolSize	**REFPluginConfiguration**	The initial size of the database connection pool. The default is **1**.
minPoolSize	**REFPluginConfiguration**	The minimum number of connections in the pool. The default is **1**.
maxPoolSize	**REFPluginConfiguration**	The maximum number of connections in the pool. The default is **50**.
connectionWaitTimeout	**REFPluginConfiguration**	The time in seconds to wait for a connection from the pool. The default is **5**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
validateConnectionOnBorrow	**REFPluginConfiguration**	Whether to validate connections on borrow. The default is **false**.
sqlForValidateConnection	**REFPluginConfiguration**	The SQL query used to validate a connection. The default is **SELECT SYSDATE FROM DUAL**.
fastConnectionFailoverEnabled	**REFPluginConfiguration**	Whether fast connection failover is enabled. The default is **false**.
onsConfiguration	**REFPluginConfiguration**	The Oracle Notification Service (ONS) configuration specifies the ONS settings.
maxStatements	**REFPluginConfiguration**	The maximum number of statements allowed in the pool. The default is **50**.
jmxEnabled	**REFPluginConfiguration**	Whether JMX monitoring is enabled. The default is **true**.
metricUpdateInterval	**REFPluginConfiguration**	The interval in seconds for updating metrics. The default is **10**.
poolLogLevel	**REFPluginConfiguration**	The log level for connection pool messages. The default is **INFO**.
control_file_dir	**REFPluginConfiguration**	The directory for control files. The default is **/home/charging/rem_input/**.
load_thread_capacity	**REFPluginConfiguration**	The number of threads for loading operations. The default is **30**.
update_thread_capacity	**REFPluginConfiguration**	The number of threads for update operations. The default is **20**.
concurrent_updaters	**REFPluginConfiguration**	The number of concurrent updaters. The default is **4**.
updater_backlog_limit	**REFPluginConfiguration**	The backlog limit for updaters. The default is **50**.
zip_buffer_size	**REFPluginConfiguration**	The zip input buffer size (in bytes) used for BLOB insertions in the ZIP_DB operation mode. The default is **8192**.
refmode	**REFPluginConfiguration**	The reference mode. The default is **DIRECT**.
creation_process	**REFPluginConfiguration**	The process used for creation. The default is **RATING_PIPELINE**.
successmode	**REFPluginConfiguration**	The success mode for operations. The default is **delete**.
dbSSLEnabled	**REFPluginConfiguration**	Whether SSL for database connection is enabled (set to "true" to enable). The default is **false**.
dbSSLClientAuth	**REFPluginConfiguration**	Whether client authentication is required for SSL/TLS connection. The default is **false**.
sslTrustStoreLocation	**REFPluginConfiguration**	The file path to the SSL/TLS truststore location for database connection. The default is **/scratch/db_wallet/cwallet.sso**.
sslKeyStoreLocation	**REFPluginConfiguration**	The file path to the SSL/TLS keystore location for database connection. The default is **/scratch/db_wallet/cwallet.sso**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
sslTrustStorePwd	**REFPluginConfiguration**	The password for accessing the SSL/TLS truststore. The default is **infranet.rel.db.ssl.truststore.password**.
sslKeyStorePwd	**REFPluginConfiguration**	The password for accessing the SSL/TLS keystore. The default is **infranet.rel.db.ssl.keystore.password**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
eventStream.*	**REFPluginConfiguration**	The eventStream property is effective when refMode is STREAM. In STREAM mode, producer.bootstrap_servers must be set to a valid Kafka broker URL. • **batch_size**: The batch size (0 = unlimited, 1 = one event per message). The default is **0**. • **compression**: Whether messages are compressed or not (NONE, ZIP, GZIP). The default is **GZIP**. • **format**: The what format is used to represent RatedEvent records (JSONFILE, ZIPFILE, CUSTOM). The default is **ZIPFILE**. • **custom_class**: The an optional class that converts RatedEvent objects into stream payloads. The default is **com.oracle.brm.ref_utils.EventToCustomPayloadConverterSample**. • **custom_csv**: Whether custom CSV output is enabled. The default is **true**. • **schema**: The what schema is used for the message format (NONE, CLOUDEVENTS). The default is **NONE**. • **use_snake_case**: Whether JSON node keys use snake case instead of camel case. The default is **true**. • **topic_1**: The topic configuration. The default is **RatedEvents-1**. • **topic_2**: The topic 2 setting specifies the configured value. The default is **RatedEvents-2**. • **topic**: The topic setting specifies the configured value. The default is **RatedEvents**. • **producer.bootstrap_servers**: The Kafka producer settings for STREAM mode. • **producer.enable_idempotence**: Whether idempotent Kafka production is enabled. The default is **true**. • **producer.acks**: The acks setting specifies the configured value. The default is **all**. • **producer.retries**: The retries setting specifies the configured value. The default is **3**. • **producer.max_in_flight_requests_per_connection**: The max in flight requests per connection setting specifies the configured value. The default is **1**. • **producer.client_id**: The client ID setting specifies the configured value. The default is **REF-KAFKA**. • **producer.key_serializer**: The key serializer setting specifies the configured value. The default is **org.apache.kafka.common.serialization.StringSerializer**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
		• **producer.value_serializer**: The value serializer setting specifies the configured value. The default is **org.apache.kafka.common.serialization.ByteArraySerializer**. • **producer.reconnect_backoff_ms**: The reconnect backoff ms setting specifies the configured value. The default is **2000**. • **producer.reconnect_backoff_max_ms**: The reconnect backoff max ms setting specifies the configured value. The default is **10000**. • **producer.max_request_size**: The max request size setting specifies the configured value. The default is **1048576**. • **schemas.cloudevents.config.specversion**: The specversion setting specifies the configured value. The default is **1.0**. • **schemas.cloudevents.config.type**: The type setting specifies the configured value. The default is **com.oracle.brm.event**. • **schemas.cloudevents.config.source**: The source setting specifies the configured value. The default is **com.oracle.ece**. • **schemas.cloudevents.config.extensions.extension1**: The extension1 setting specifies the configured value. The default is **extension-1-value**. • **schemas.cloudevents.config.extensions.extension2**: The extension2 setting specifies the configured value. The default is **extension-2-value**. • **schemas.cloudevents.headers.h1**: The h1 setting specifies the configured value. The default is **CE1**. • **schemas.cloudevents.headers.h2**: The h2 setting specifies the configured value. The default is **CE2**. • **schemas.none.config.dummy**: The dummy setting specifies the configured value. The default is **dummy**. • **schemas.none.headers.h1**: The h1 setting specifies the configured value. The default is **N1**. • **schemas.none.headers.h2**: The h2 setting specifies the configured value. The default is **N2**.
cmService.*	N/A	The details for the CM service: • **ipaddress**: The IP address for the CM service. • **port**: The port for CM service communication. The default is **11960**. • **targetPort**: The target port for routing traffic. The default is **27015**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
brmgatewayService.port	N/A	The port setting specifies the service port. The default is **15502**.
cdrgatewayService.port	N/A	The port setting specifies the service port. The default is **8084**.
diametergatewayService.*	N/A	The port details for the Diameter Gateway service: • **port**: The port setting specifies the service port. The default is **3868**. • **type**: The type setting specifies the service type. The default is **NodePort**.
radiusgatewayService.*	N/A	The port details for the RADIUS Gateway service: • **port**: The port setting specifies the service port. The default is **1812**. • **type**: The type setting specifies the service type. The default is **NodePort**.
httpgatewayService.*	N/A	The port details for the HTTP Gateway service: • **port**: The port setting specifies the service port. The default is **8080**. • **type**: The type setting specifies the service type. The default is **NodePort**.
jmxservice.*	N/A	The port details for the JMX service: • **port**: The port setting specifies the service port. The default is an empty value. • **exposeExternally**: Whether external JMX service exposure is enabled. The default is **true**. • **type**: The type of the service Used only when exposeExternally is explicitly enabled. The default is **ClusterIP**.
type	**monitoringagent**	The NodePort service type allows access to application from outside. The default is **NodePort**.
logs.*	**pvc**	The details for logs: • **name**: The name for the ECE log files. • **accessModes**: The access mode for the PVC. The default is **ReadWriteMany**. • **storage**: The storage space required initially to create this PVC. If the storage specified here is not available in the machine, ensure that the PVC is not created and that the pods do not get initialized. The default is **500Mi**. • **createOption.***: The by default, the PVC uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See " Using Static Volumes " in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-1 (Cont.) Elastic Charging Engine Keys

Key	Path in values.yaml File	Description
brmconfig.*	**pvc**	The details for brmconfig: • **name**: The name of the BRM Config PVC, in which all BRM configuration files such as the payload configuration file are exposed outside of the pod. The default is **brmconfig-pvc**. • **accessModes**: The access mode for the PVC. The default is **ReadWriteOncePod**. • **storage**: The storage space required initially to create this PVC. If the storage specified here is not available in the machine, ensure that the PVC is not created and that the pods do not get initialized. The default is **500Mi**. • **createOption.***: The by default, the PVC uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See " Using Static Volumes " in *BRM Cloud Native System Administrator's Guide*.
cdrformatter.*	**pvc**	The details for cdrformatter: • **name**: The name for the CDR formatter PVC. The default is **cdrformatter-pvc**. • **accessModes**: The access mode for the PVC. The default is **ReadWriteMany**. • **storage**: The storage space required initially to create this PVC. If the storage specified here is not available in the machine, ensure that the PVC is not created and that the pods do not get initialized. The default is **500Mi**. • **createOption.***: The by default, the PVC uses dynamic volumes. To use a static volume instead, you must add the **createOption** key. See " Using Static Volumes " in *BRM Cloud Native System Administrator's Guide*.
storageClass.name	N/A	The name of the storage class for dynamic volume provisioning.
coherenceClusterRackSafe.*	N/A	The settings for making the ECE Coherence cluster rack-safe. • **enabled**: Whether to make the ECE Coherence cluster rack-safe (**true**) or not (**false**). The default is **false**. If set to **true**, you must map worker nodes to racks by using the **workerNodeRackNameMapping** key. • **workerNodeRackNameMapping**: The mapping of worker node IP addresses or names to rack names. Prefix each worker node IP address or name with **node_** and specify its rack name in the format **node_WorkerNode: RackName**. For OCI deployments, you can use the fault domain hosting the worker node as the rack name.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Configuring Gateway Service Types and NodePorts

The ECE Helm chart creates Kubernetes services for Diameter Gateway, RADIUS Gateway, and HTTP Gateway. By default, these gateway services use the **NodePort** service type.

You can change the service type in the **override-values.yaml** file for **oc-cn-ece-helm-chart** when your Kubernetes environment exposes gateway traffic through another method, such as an ingress controller or a load balancer.

To configure the gateway service types:

1. Open the **override-values.yaml** file for **oc-cn-ece-helm-chart**.
2. Set the **type** key under **diametergatewayService**, **radiusgatewayService**, and **httpgatewayService**.

   For example:

   ```
   diametergatewayService:
     port: 3868
     type: NodePort

   radiusgatewayService:
     port: 1812
     type: NodePort

   httpgatewayService:
     port: 8080
     type: NodePort
   ```

   You can set **type** to **ClusterIP**, **NodePort**, or **LoadBalancer**, depending on how gateway traffic is exposed in your Kubernetes environment.
3. If the service type is **NodePort**, set the **NodePort** values for the gateway service instances.

   For example:

   ```
   diametergateway:
     diametergatewayList:
       - coherenceMemberName: "diametergateway1"
         diameterServiceNodePort: 31851
       - coherenceMemberName: "diametergateway2"
         diameterServiceNodePort: 32050

   radiusgateway:
     radiusgatewayList:
       - coherenceMemberName: "radiusgateway1"
         radiusServiceNodePort: 30182
       - coherenceMemberName: "radiusgateway2"
         radiusServiceNodePort: 31135

   httpgateway:
         serverHttpExternalPort: 31500
   ```

   For multiple Diameter Gateway or RADIUS Gateway instances, specify the NodePort value under each instance in the corresponding gateway list.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> The **diameterServiceNodePort**, **radiusServiceNodePort**, and **serverHttpExternalPort** keys are used only when the corresponding gateway service **type** is set to **NodePort**. If you do not specify these keys, Kubernetes assigns **NodePort** values automatically.

4. Run the **helm upgrade** command.
5. Verify the gateway service type and port configuration:

   ```
   kubectl get svc
   ```

   When the gateway services are configured as **NodePort**, the output shows the configured **NodePort** values.

   For example:

   ```
   NAME          TYPE       PORT(S)
   ece-dgw-1     NodePort   3868:31851/TCP
   ece-dgw-2     NodePort   3868:32050/TCP
   ece-http      NodePort   8080:31500/TCP
   ece-rgw       NodePort   1812:30182/UDP
   ece-rgw-2     NodePort   1812:31135/UDP
   ```

   When the gateway services are configured as **ClusterIP**, the output does not show **NodePort** values.

   For example:

   ```
   NAME          TYPE        PORT(S)
   ece-dgw-1     ClusterIP   3868/TCP
   ece-dgw-2     ClusterIP   3868/TCP
   ece-http      ClusterIP   8080/TCP
   ece-rgw       ClusterIP   1812/UDP
   ece-rgw-2     ClusterIP   1812/UDP
   ```

Enabling SSL in Elastic Charging Engine

> **Note**
>
> For more information about securing communications between ECE and external applications, see "Securing ECE Communication" in *BRM Cloud Native System Administrator's Guide*.

To complete the configuration for SSL setup in ECE:

1. Configure the SSL KeyStores by doing one of the following:
   - Generate the Identity and Trust KeyStores and then move your files, such as **identity.p12** and **trust.p12**, under the **oc-cn-ece-helm-chart/ece/keystore** directory.
   - Pre-create the Kubernetes Secret for the Identity and Trust KeyStore files and set the **secretEnv.extECESecret** key in your **override-values.yaml** file for both **oc-cn-ece-helm-chart**.

     For more information, see "Managing KeyStore Certificates and Wallets" in *BRM Cloud Native System Administrator's Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Set these keys in the **override-values.yaml** file for **oc-cn-ece-helm-chart**:
   - **sslconnectioncertificates.SSLENABLED**: Set this to **true**.
   - **sslEnabled**: Set this to **true** in **emGatewayConfigurations**, **httpGatewayConfigurations**, and **BRMConnectionConfiguration**.
   - **migration.pricingUpdater.keyStoreLocation**: Set this to **/home/charging/opt/ECE/oceceserver/config/client.jks**.
   - **charging.brmWalletServerLocation**: Set this to the name of the BRM server wallet file, such as **cwallet.sso**.
   - **charging.emGatewayConfigurations.emGatewayConfigurationList.emGateway1Config.wallet**: Set this to the BRM wallet location.
   - **charging.emGatewayConfigurations.emGatewayConfigurationList.emGateway2Config.wallet**: Set this to the BRM wallet location.
   - **charging.radiusGatewayConfigurations.wallet**: Set this to the BRM wallet location.
   - **charging.connectionConfigurations.BRMConnectionConfiguration.brmwallet**: Set this to the BRM wallet location.
3. Deploy your ECE cloud native services by following the instructions in "Deploying BRM Cloud Native Services".

Enabling SSL for the HTTP Gateway Server

When you enable SSL for the HTTP Gateway server, the ECE Helm chart does not automatically generate the HTTP Gateway server certificate or server SSL keystore.

Before installing or updating the ECE Helm chart:

1. Create or obtain a private key and server certificate for the HTTP Gateway server.
2. Create a PKCS12 keystore containing the private key and server certificate.
3. Configure the HTTP Gateway server SSL keystore by doing one of the following:
   - To store the keystore in the ECE Helm chart:
     a. Move the PKCS12 keystore file to the **oc-cn-ece-helm-chart/secrets/httpgateway** directory.
     b. In your **override-values.yaml** file for **oc-cn-ece-helm-chart**, set the following keys:
        - **httpgateway.serverSslKeyStore:** Set this to the name of the PKCS12 keystore file.
        - **createWalletsAsSecrets:** Set this to **true**.
   - To use an external Kubernetes secret:
     a. Create a Kubernetes secret containing the HTTP Gateway server PKCS12 keystore.
     b. In your **override-values.yaml** file **oc-cn-ece-helm-chart**, set **httpgateway.extServerSSLKeyStoreSecret** to the name of the Kubernetes secret.
4. In your **override-values.yaml** file, set the following keys:
   - **httpgateway.serverSslKeyStoreType**: Set this to **PKCS12**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **httpgateway.serverSslKeyStoreAlias**: Set this to the private key alias in the keystore. The default is **oracle**.
- **secretEnv.ECEHTTPGATEWAYSERVERSSLKEYSTOREPASSWORD**: Set this to the password for the PKCS12 keystore.

5. For each applicable HTTP Gateway instance, set the **httpgateway.httpgatewayList[].httpGatewayConfiguration.serverSslEnabled** key to **true**.
6. Deploy the ECE Helm chart.

Connecting ECE Cloud Native to an SSL-Enabled Database

ECE uses Kubernetes secrets to provide SSL wallet and truststore files that secure connections to the ECE persistence database and the BRM database.

> **Note**
>
> In earlier ECE releases, the ECE wallet was stored on the shared **ece-wallet-pvc**. This PVC is no longer used or created in current ECE releases. ECE wallet files are provisioned from Kubernetes secrets into pod-local storage.

You can provide the SSL files in either of the following ways:

- Add the SSL files to the ECE Helm chart and allow the chart to create the required Kubernetes secrets during deployment.
- Create Kubernetes secrets before you deploy ECE and configure ECE to use those secrets.

To connect ECE cloud native services to an SSL-enabled Oracle database:

1. Provide the SSL wallet files in one of the following ways:
   - Using the ECE Helm Chart to Create Kubernetes Secrets
   - Using Pre-created Kubernetes Secrets
2. Configure the ECE persistence database connection. Under **charging.connectionConfigurations.OraclePersistenceConnectionConfigurations**, configure the following keys:
   - **dbSSLEnabled**: Set this to **true**.
   - **dbSSLType**: Set this to the required SSL connection type: **oneway**, **twoway**, or **none**.
   - **sslServerCertDN**: Set this to the SSL server certificate DN. The default is **DC=local,DC=oracle,CN=pindb**.
   - **trustStoreLocation**: Specify the truststore file name when the ECE Helm chart creates the secret. If you use a pre-created Kubernetes secret, specify the key in the secret that contains the truststore.
   - **trustStoreType**: Set this to the truststore file type: **SSO** or **PKCS12**.
   - **extPersistenceDBSSLWalletSecret**: If you use a pre-created Kubernetes secret, set this to the name of the secret that contains the ECE database SSL truststore.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3. Configure Customer Updater for an SSL-enabled Oracle AQ database queue. Under **customerUpdater.customerUpdaterList.oracleQueueConnectionConfiguration**, configure the following keys:
   - **dbSSLEnabled**: Set this to **true**.
   - **dbSSLType**: Set this to the required SSL connection type: **oneway**, **twoway**, or **none**.
   - **sslServerCertDN**: Set this to the SSL server certificate DN. The default is **DC=local,DC=oracle,CN=pindb**.
   - **trustStoreLocation**: Specify the truststore file name when the ECE Helm chart creates the secret. If you use a pre-created Kubernetes secret, specify the key in the secret that contains the truststore.
   - **trustStoreType**: Set this to the truststore file type: **SSO** or **PKCS12**.
   - **extBRMDBSSLWalletSecret**: If you use a pre-created Kubernetes secret, set this to the name of the secret that contains the BRM database SSL truststore.

   > **Note**
   >
   > For database connectivity, ECE supports the database service name, but not the database service ID. Set the following keys to the database service name:
   >
   > - **charging.connectionConfigurations.OraclePersistenceConnectionConfigurations.sid**
   > - **customerUpdater.customerUpdaterList.oracleQueueConnectionConfiguration.sid**

4. Configure the Oracle Net files for the ECE and BRM database connections.
   a. Obtain the **tnsnames.ora** and **sqlnet.ora** files for the SSL-enabled ECE and BRM databases.
   b. In the ECE database **sqlnet.ora** file, configure the wallet location as follows:

```
WALLET_LOCATION =
   (SOURCE =
      (METHOD = FILE)
      (METHOD_DATA =
         (DIRECTORY = /home/charging/opt/ECE/ocecеserver/config/
ece_ssl_db_wallet/schema1)
      )
   )
```

   c. In the BRM database **sqlnet.ora** file, configure the wallet location as follows:

```
WALLET_LOCATION =
   (SOURCE =
      (METHOD = FILE)
      (METHOD_DATA =
         (DIRECTORY = /home/charging/opt/ECE/ocecеserver/config/
brm_ssl_db_wallet/schema1)
      )
   )
```

   d. When you build your customized ECE image:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Add the ECE Oracle Net files to **/home/charging/ext/ora_files/ece**.
- Add the BRM Oracle Net files to **/home/charging/ext/ora_files/brm**.

e. Build and publish the customized ECE image, and configure your ECE deployment to use that image.

5. Configure SSL files for other ECE integrations as required.

   Use the corresponding Kubernetes secret settings when you configure SSL for other ECE components. For example:

```
charging:
  kafkaConfigurations:
    kafkaConfigurationList:
      - kafkaTrustStoreLocation: certificate_key
        extKafkaTrustStoreSecret: external_secret_name

migration:
  pricingUpdater:
    keyStoreLocation: certificate_key
    extPDCKeyStoreSecret: external_secret_name

JMSConfiguration:
  NotificationQueue:
    - KeyStoreLocation: certificate_key
      extJMSKeyStoreSecret: external_secret_name

  BRMGatewayNotificationQueue:
    - KeyStoreLocation: certificate_key
      extJMSKeyStoreSecret: external_secret_name

  DiameterGatewayNotificationQueue:
    - KeyStoreLocation: certificate_key
      extJMSKeyStoreSecret: external_secret_name
```

   For information about the available configuration keys, see "Adding Elastic Charging Engine Keys".

Using the ECE Helm Chart to Create Kubernetes Secrets

To have the ECE Helm chart create the Kubernetes secrets:

1. In your **override-values.yaml** file for **oc-cn-ece-helm-chart**, set **createWalletsAsSecrets** to **true**.
2. Place the ECE database SSL wallet files in the **oc-cn-ece-helm-chart/secrets/db_wallets/ece_ssl_db_wallet/schema***n* directory, where *n* is the schema number.
3. Place the BRM database SSL wallet files in the **oc-cn-ece-helm-chart/secrets/db_wallets/brm_ssl_db_wallet/schema***n* directory, where *n* is the schema number.
4. For a multischema system, create a directory for each schema and add the appropriate wallet files to each directory.
5. From the **oc-cn-ece-helm-chart** directory, grant the required permissions to the secrets directory:

```
chmod -R 775 secrets
```

Using Pre-created Kubernetes Secrets

To use pre-created Kubernetes secrets:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Create a Kubernetes secret that contains the SSL truststore for the ECE persistence database.
2. Create a Kubernetes secret that contains the SSL truststore for the BRM database connection.
3. Open your **override-values.yaml** file for **oc-cn-ece-helm-chart**.
4. Under **charging.connectionConfigurations.OraclePersistenceConnectionConfigurations**, set **extPersistenceDBSSLWalletSecret** to the name of the ECE Kubernetes secret.
5. Set **trustStoreLocation** to the key in the secret that contains the truststore for the ECE persistence database.
6. Under **customerUpdater.customerUpdaterList.oracleQueueConnectionConfiguration**, set **extBRMDBSSLWalletSecret** to the name of that BRM Kubernetes secret.
7. Set **trustStoreLocation** to the key in the secret that contains the truststore for the BRM database connection.

ECE provides **extPersistenceDBSSLWalletSecret** and **extBRMDBSSLWalletSecret** for externally supplied Kubernetes Secrets.

About Elastic Charging Engine Volume Mounts

ECE uses persistent volumes for data that requires persistence or shared access across pods. Use RWX storage only for workloads that require shared writable storage.

Table 10-2 summarizes the ECE storage model.

Table 10-2 ECE Storage Model Details

Storage	Storage Model	Description
logs-pvc	Optional RWX PVC or pod-local storage	ECE uses **logs-pvc** when you configure **pvc.logs.name**. If you do not configure a logs PVC, ECE uses pod-local **emptyDir** storage and continues to write supported log output to the console. Files stored only in **emptyDir** do not persist when Kubernetes deletes or re-creates the pod. Utilities such as **query.sh**, subscriber tracing, GC logging, and scripts that write directly to the logs directory can continue to create log files, but those files are not persistent.
brmconfig-pvc	**ReadWriteOncePod**	ECE uses **brmconfig-pvc** as a single-writer persistent volume for BRM configuration data generated by Pricing Updater. Keep Pricing Updater configured with a single replica when you use this storage model.
cdrformatter-pvc	RWX PVC	ECE uses shared writable storage for CDR Formatter output because the generated files must be available in a common shared location.
SDK storage	Pod-local	ECE provides the SDK content in the container at **/home/charging/opt/ECE/ocecesdk**. Files that SDK jobs generate remain pod-local unless you copy them to persistent storage.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 10-2 (Cont.) ECE Storage Model Details

Storage	Storage Model	Description
/home/charging/ext	Image-backed	Use image layering to provide supported non-SSL files under **/home/charging/ext**, including third-party JAR files, custom ECE data, Diameter configuration, Oracle Net configuration files, and subscriber trace configuration. Use Kubernetes secrets for SSL wallets, keystores, and other supported sensitive data.
ECE wallet	Pod-local **emptyDir**	Each wallet-using pod creates its wallet in pod-local **emptyDir** storage under **/home/charging/wallet** from mounted Kubernetes secrets. For long-running wallet-using pods, the ECE wallet sidecar monitors mapped Kubernetes secret values and updates the corresponding wallet entries while the pod is running. When Kubernetes re-creates a pod, ECE re-creates the wallet from the current secret values. Do not configure an ECE wallet PVC. For information about rotating persistence database credentials at runtime, see "Rotating Persistence Database Passwords" in *BRM Cloud Native System Administrator's Guide*.

For the list of third-party libraries to download, see *BRM Compatibility Matrix*.

Configure the supported persistent-volume settings in the **override-values.yaml** file for **oc-cn-ece-helm-chart**. For information about the available storage configuration keys, see "Adding Elastic Charging Engine Keys".

Loading Custom Diameter AVP

Use image layering to add custom Diameter AVP files to your ECE cloud native deployment. ECE reads custom Diameter AVP files from **/home/charging/ext/diameter** in the ECE image.

To load custom Diameter AVPs into your ECE cloud native environment:

1. In the files that you use to build your customized ECE image, create the **diameter** directory under **/home/charging/ext**.
2. Add the custom AVP file, such as **dictionary_custom.xml**, to **/home/charging/ext/diameter**.
3. Build and publish the customized ECE image.
4. Configure your ECE deployment to use the customized image.
5. Run the **helm install** or **helm upgrade** command for the ECE Helm release.

If you add or update a custom AVP after ECE is deployed, restart the affected Diameter Gateway pod:

1. In your **override-values.yaml** file for **oc-cn-ece-helm-chart**, increment the **restartCount** value for the applicable entry in **diametergateway.diametergatewayList**.
2. Run the **helm upgrade** command to update the release.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Generating CDRs for Unrated Events

By default, the httpgateway pod sends all 5G usage requests to the ecs pod for online and offline charging.

You can configure httpgateway to convert some 5G usage requests into call detail record (CDR) files based on the charging type. You can then send the CDR files to roaming partners, a data warehousing system, or legacy billing systems for rating. For more information, see "About Generating CDRs" in *ECE Implementing Charging*.

You use the following to generate CDRs:

- httpgateway pod
- cdrgateway pod
- cdrFormatter pod
- CDR database

The cdrgateway and cdrFormatter pods can be scaled together, with one each per schema, or independently of the schemas. For more information, see "Scaling the cdrgateway and cdrFormatter Pods".

For details about the CDR format, see "CHF-CDR Format" in *ECE 5G CHF Protocol Implementation Conformance Statement*.

To set up ECE cloud native to generate CDRs:

1. Configure your httpgateway pod to do the following:
   - Generate CDRs (set **cdrGenerationEnabled** to **true**).
   - Route offline charging requests to the ecs pod for rating (set **rateOfflineCDRinRealtime** to **true**) or to the cdrgateway pod for generating CDRs (set **rateOfflineCDRinRealtime** to **false**).
   - Route online charging requests to the ecs pod for rating (set **generateCDRsForOnlineRequests** to **false**) or to the cdrgateway pod for generating CDRs (set **generateCDRsForOnlineRequests** to **true**).
2. Configure the cdrgateway pod to connect to the CDR database and do the following:
   - Generate individual CDR records for each request (set **individualCdr** to **true**) or aggregate multiple requests into a CDR record based on trigger criteria (set **individualCdr** to **false**). For information about the trigger criteria, see "About Trigger Types" in *ECE Implementing Charging*.
   - Store CDR records in an Oracle NoSQL database (set **isNoSQLConnection** to **true**) or in an Oracle database (set **isNoSQLConnection** to **false**).
3. Configure the cdrFormatter pod to do the following:
   - Retrieve batches of CDR records from the CDR database and pass them to a specified cdrFormatter plug-in for processing.
   - Purge processed CDR records from the CDR database older than a specified number of days (configured in **retainDuration**).
   - Purge orphan CDR records from the CDR database.

     Orphan CDR records are incomplete ones that are older than a specified number of seconds (configured in **cdrOrphanRecordCleanupAgeInSec**). Orphan CDR records can be created when your ECE system goes down due to maintenance or failure.

Copyright © 2019, 2026, Oracle and/or its affiliates.

4. Configure the cdrFormatter plug-in to do the following:
   - Write a specified number of CDR records to each CDR file (set **maxCdrCount** to the maximum number).
   - Create JSON-formatted CDR files and then store them in your file system (set **enableDiskPersistence** to **true**) or send them to your Kafka messaging service (set **enableKafkaIntegration** to **true**).

To generate CDRs in ECE cloud native, you configure the following entries in your **override-values.yaml** file. This example configures:

- httpgateway to route both online and offline charging requests to cdrgateway.
- cdrgateway to aggregate multiple requests into a CDR record and then store it in an Oracle NoSQL database.
- cdrFormatter to retrieve CDR records in batches of 2500 from the Oracle NoSQL database and then send them to the default plug-in module. Immediately after CDR records are retrieved, cdrFormatter purges them from the database. It would also purge orphan records older than 200 seconds from the database.
- The cdrFormatter plug-in to create CDR files with a maximum of 20000 CDR records and an **.out** file name extension. It would store them in your file system in the path **/home/charging/cdr_input**.

```
cdrFormatter:
 cdrFormatterList:
   - schemaNumber: "1"
     replicas: 1
     coherenceMemberName: "cdrformatter1"
     jmxEnabled: true
     jvmGCOpts: "-XX:+UnlockExperimentalVMOptions  -XX:+AlwaysPreTouch -
XX:G1RSetRegionEntries=2048 -XX:ParallelGCThreads=10 -XX:+ParallelRefProcEnabled    -
XX:MetaspaceSize=100M -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -
XX:+PrintTenuringDistribution -XX:+PrintAdaptiveSizePolicy -XX:-UseGCLogFileRotation -
XX:+UseG1GC -XX:NumberOfGCLogFiles=99"
     jvmOpts: "-Xms16g -Xmx20g -Dece.metrics.http.service.enabled=true"
     cdrFormatterConfiguration:
       name: "cdrformatter1"
       clusterName: "BRM"
       primaryInstanceName: "cdrformatter1"
       partition: "1"
       isNoSQLConnection: "true"
       noSQLConnectionName: "noSQLConnection"
       connectionName: "oraclePersistence1brm"
       threadPoolSize: "6"
       retainDuration: "0"
       ripeDuration: "60"
       checkPointInterval: "6"
       maxPersistenceCatchupTime: "0"
       pluginPath: "ece-cdrformatter.jar"
       pluginType:
"oracle.communication.brm.charging.cdr.formatterplugin.internal.SampleCdrFormatterCustomP
lugin"
       pluginName: "cdrFormatterPlugin1"
       noSQLBatchSize: "2500"
       cdrStoreFetchSize: "2500"
       cdrOrphanRecordCleanupAgeInSec: "200"
       cdrOrphanRecordCleanupSleepIntervalInSec: "200"
       enableIncompleteCdrDetection: "false"

cdrgateway:
  cdrgatewayList:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    - coherenceMemberName: "cdrgateway1"
      replicas: 6
      jmxEnabled: true
      jvmGCOpts: "-XX:+UnlockExperimentalVMOptions  -XX:+AlwaysPreTouch -
XX:G1RSetRegionEntries=2048 -XX:ParallelGCThreads=10 -XX:+ParallelRefProcEnabled    -
XX:MetaspaceSize=100M -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -
XX:+PrintTenuringDistribution -XX:+PrintAdaptiveSizePolicy -XX:-UseGCLogFileRotation -
XX:+UseG1GC -XX:NumberOfGCLogFiles=99"
      jvmJMXOpts: "-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.authenticate=true -Dcom.sun.management.jmxremote.ssl=true -
Dcom.sun.management.jmxremote.local.only=true"
      jvmCoherenceOpts: "-Dpof.config=charging-pof-config.xml -
Dcoherence.override=charging-coherence-override-dev.xml  -Dcoherence.security=false -
Dsecure.access.name=admin"
      jvmOpts: "-Xms6g -Xmx8g -Dece.metrics.http.service.enabled=true -
DcdrServerCorePoolSize=64 -Dserver.sockets.metrics.bind-address=0.0.0.0 -
Dece.metrics.http.port=19612"
      restartCount: "0"
      cdrGatewayConfiguration:
        name: "cdrgateway1"
        clusterName: "BRM"
        primaryInstanceName: "cdrgateway1"
        schemaNumber: "1"
        isNoSQLConnection: "true"
        noSQLConnectionName: "noSQLConnection"
        connectionName: "oraclePersistence1"
        cdrPort: "8084"
        cdrHost: "ece-cdrgatewayservice"
        individualCdr: "false"
        cdrServerCorePoolSize: "32"
        cdrServerMaxPoolSize: "256"
        enableIncompleteCdrDetection: "false"
        retransmissionDuplicateDetectionEnabled: "false"

httpgateway:
   cdrGenerationEnabled: "true"
   cdrGenerationStandaloneMode: "true"
   rateOfflineCDRinRealtime: "false"
   generateCDRsForOnlineRequests: "true"
   httpgatewayList:
      - coherenceMemberName: "httpgateway1"
        replicas: 8
        maxreplicas: 8
        jvmGCOpts: "-XX:+AlwaysPreTouch -XX:G1RSetRegionEntries=2048 -
XX:ParallelGCThreads=10 -XX:+ParallelRefProcEnabled    -XX:MetaspaceSize=100M -
XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -
XX:+PrintTenuringDistribution -XX:+PrintAdaptiveSizePolicy -XX:-UseGCLogFileRotation -
XX:+UseG1GC -XX:NumberOfGCLogFiles=99"
        jvmOpts: "-Xms10g -Xmx14g -Djava.net.preferIPv4Addresses=true -
Dece.metrics.http.service.enabled=true -Dserver.sockets.metrics.bind-address=0.0.0.0 -
Dece.metrics.http.port=19612"
        httpGatewayConfiguration:
           name: "httpgateway1"
           processingThreadPoolSize: "200"
           processingQueueSize: "32768"
           kafkaBatchSize: "10"

charging:
  connectionConfigurations:
    OraclePersistenceConnectionConfigurations:
      - retryCount: "1"
        retryInterval: "1"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
        maxStmtCacheSize: "100"
        connectionWaitTimeout: "3000"
        timeoutConnectionCheckInterval: "3000"
        inactiveConnectionTimeout: "3000"
        databaseConnectionTimeout: "6000"
        persistenceInitialPoolSize: "4"
        persistenceMinPoolSize: "4"
        persistenceMaxPoolSize: "20"
        reloadInitialPoolSize: "0"
        reloadMinPoolSize: "0"
        reloadMaxPoolSize: "20"
        ratedEventFormatterInitialPoolSize: "6"
        ratedEventFormatterMinPoolSize: "6"
        ratedEventFormatterMaxPoolSize: "24"

  cdrFormatterPlugins:
    cdrFormatterPluginConfigurationList:
      cdrFormatterPluginConfiguration:
        name: "cdrFormatterPlugin1"
        tempDirectoryPath: "/tmp/tmp"
        doneDirectoryPath: "/home/charging/cdr_input"
        doneFileExtension: ".out"
        enableKafkaIntegration: "false"
        enableDiskPersistence: "true"
        maxCdrCount: "20000"
        staleSessionCauseForRecordClosingString: "PARTIAL_RECORD"
        enableStaleSessionCleanupCustomField: "false"
```

Scaling the cdrgateway and cdrFormatter Pods

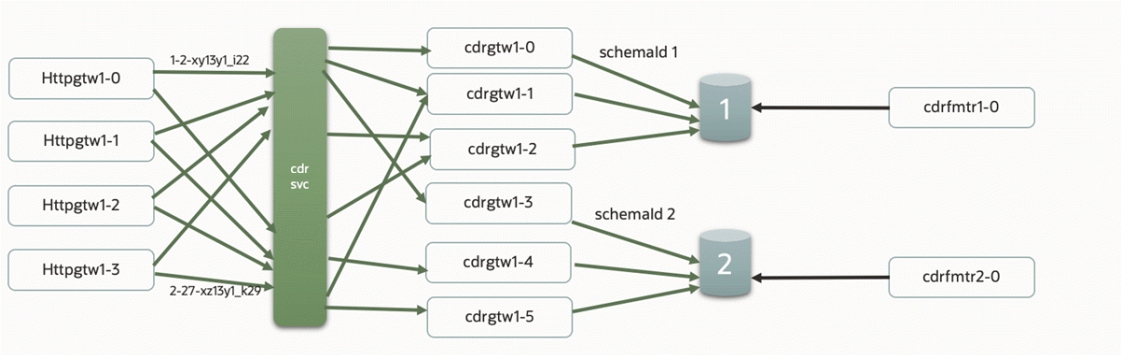
To increase performance and throughput, you can scale the cdrgateway and cdrFormatter pods together, with one each per schema, or scale them independently of the schemas.

Figure 10-1 shows an example of scaled cdrgateway and cdrFormatter pods that have CDR storage in an Oracle Database. This example contains:

- One cdrgateway multi-replica deployment for all ECE schemas. All cdrgateway replicas have a single CDR Gateway service acting as a front end to httpgateway.
- One cdrFormatter single-replica deployment for each ECE schema. Each cdrFormatter reads persisted CDRs from its associated ECE schema.

httpgateway forwards CDR requests to cdrgateway replicas in round-robin fashion. In this example, cdrgateway replicas 1-0, 1-1, and 1-2 persist CDRs in schema 1 tables, and replicas 1-3, 1-4, and 1-5 persist CDRs in schema 2 tables.

Figure 10-1 Scaled Architecture with an Oracle Database

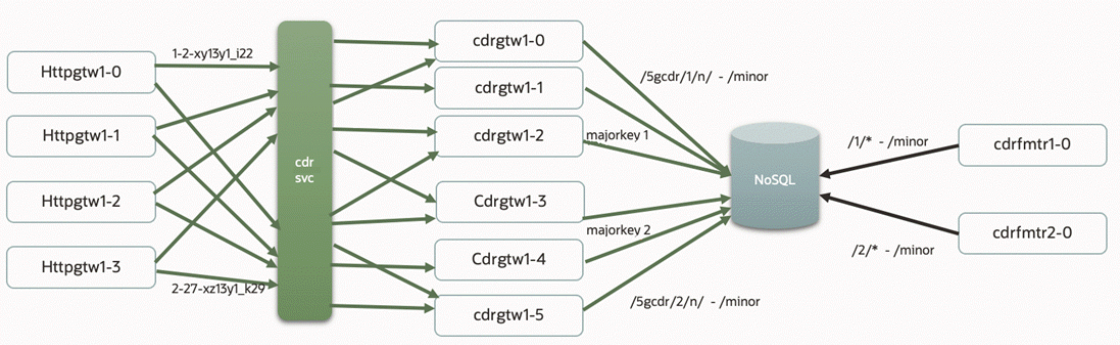


Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 10-2 shows an example of scaled cdrgateway and cdrFormatter pods that have CDR storage in an Oracle NoSQL Database. This example contains:

- One cdrgateway multi-replica deployment for all ECE schemas. All cdrgateway replicas have a single CDR Gateway service acting as a front end to the httpgateway.
- One cdrFormatter single-replica deployment for each major key partition in the ECE schema. Each cdrFormatter reads persisted CDRs from its associated partition.

Figure 10-2 Scaled Architecture with a NoSQL Database



Configuring ECE to Support Prepaid Usage Overage

You can configure ECE cloud native to capture any overage amounts by prepaid customers during an active session, which can help you prevent revenue leakage. If the network reports that the number of used units during a session is greater than a customer's available allowance, ECE cloud native charges the customer up to the available allowance. It can then create an overage record with information about the overage amount and sends it to the ECE Overage topic. You can create a custom solution for reprocessing the overage amount later on.

For example, assume a customer has a prepaid balance of 100 minutes, but uses 130 minutes during a session. ECE cloud native would charge the customer for 100 minutes, create an overage record for the remaining 30 minutes of usage, and then write the overage topic to the ECE Overage Kafka topic.

When the prepaid usage overage is disabled, ECE cloud native charges the customer for the full amount regardless of the amount of funds in the customer's balance.

To configure ECE cloud native to support prepaid usage overage, do the following:

- Ensure that ECE cloud native is connected to your Kafka Server
- Enable ECE cloud native to support prepaid usage overage
- Create an ECE Overage topic in your Kafka Server

To do so, set the following keys in your **override-values.yaml** file for **oc-cn-ece-helm-chart** and then run the **helm upgrade** command:

- **charging.kafkaConfigurations.kafkaConfigurationList.***: Specify how to connect ECE to your Kafka Server.
- **charging.server.checkReservationOverImpact**: Set this to **true**.
- **charging.kafkaConfigurations.kafkaConfigurationList.overageTopicName**: Set this to the name of the Kafka topic where ECE will publish overage records.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> If your system does not contain Kafka topics, you can configure ECE to push overage details to a separate log file instead. To do so, in your **override-values.yaml** file for **oc-cn-ece-helm-chart**, set the **charging.ecs.jvmOpts** key to **-Deceoveragelogdir=/home/charging/opt/ECE/ocecеserver/logs**.

Recording Failed ECE Usage Requests

ECE cloud native may occasionally fail to process usage requests. For example, a data usage request could fail because a customer has insufficient funds. You can configure ECE cloud native to publish details about failed usage requests, such as the user ID and request payload, to the ECE failure topic in your Kafka server. Later on, you can reprocess the usage requests or view the failure details for analysis and reporting.

To configure ECE cloud native to record failed ECE usage requests:

- Ensure that ECE cloud native is connected your Kafka Server
- Enable the recording of failed ECE usage requests
- Create an ECE failure topic in your Kafka Server

To do so, set the following keys in your **override-values.yaml** file for **oc-cn-ece-helm-chart**:

- **charging.kafkaConfigurations.kafkaConfigurationList.***: Specify how to connect ECE to your Kafka Server.
- **charging.kafkaConfigurations.kafkaConfigurationList.persistFailedRequestsToKafkaTopic**: Set this to **true**.
- **charging.kafkaConfigurations.kafkaConfigurationList.failureTopicName**: Set this to the name of the topic that stores information about failed ECE usage requests.

Loading BRM Configuration XML Files

BRM is configured by using the **pin_notify** and **payloadconfig_ece_sync.xml** files. To ensure that the BRM pod can access these files for configuring the EAI Java Server (eai_js), they are exposed through the brm_config PVC within the pricingupdater pod. When new metadata is synchronized with ECE, if there are updates to the payload configuration file, it will create a new file in the location which can be accessed and configured in BRM.

For more information, see "Enabling Real-Time Synchronization of BRM and ECE Customer Data Updates" in *ECE Implementing Charging*.

Setting Up Notification Handling in ECE

You can configure ECE cloud native to send notifications to a client application or an external application during an online charging session. For example, ECE can send a notification when a customer breaches a credit threshold or needs to request reauthorization.

You can configure ECE to send notifications by using either Apache Kafka topics or Oracle WebLogic queues:

- Creating an Apache Kafka Notification Topic
- Creating an Oracle WebLogic Notification Queue

Copyright © 2019, 2026, Oracle and/or its affiliates.

Creating an Apache Kafka Notification Topic

To create notification topics in Apache Kafka:

1. Create the following Kafka topics either in the Kafka **entrypoint.sh** script or after the Kafka pod is ready:
   - **kafka.topicName**: Set this to **ECENotifications**.
   - **kafka.suspenseTopicName**: Set this to **ECESuspenseQueue**.
2. In the ZooKeeper runtime ConfigMap, set the **ece-zookeeper-0.ece-zookeeper.***ECENameSpace***.svc.cluster.local** key to the name of the Kafka cluster.
3. Set the following Kafka and ZooKeeper environment variables:
   - **KAFKA_PORT**: Set this to the port on which Apache Kafka listens.
   - **KAFKA_HOST_NAME**: Set this to the host name of the machine on which Apache Kafka runs. If you use multiple Kafka brokers, specify a comma-separated list of host names.
   - **REPLICATION_FACTOR**: Set this to the number of topic replicas to create.
   - **PARTITIONS**: Set this to the number of Kafka partitions to create for your topics. Oracle recommends calculating the number as follows:

     [(Max Diameter Gateways * Max Peers Per Gateway) + (1 for BRM Gateway) + Internal Notifications]
   - **TOPIC_NAME**: Set this to **ECENotifications**. ECE publishes notifications to this topic.
   - **SUSPENSE_TOPIC_NAME**: Set this to **ECESuspenseQueue**. BRM publishes failed notifications to this topic for later retry.
   - **ZK_CLUSTER**: Set this to the name of your ZooKeeper cluster. This should match the value you set in step 2.
   - **ZK_CLIENT_PORT**: Set this to the port on which ZooKeeper listens for client connections.
   - **ZK_SERVER_PORT**: Set this to the ZooKeeper server port.
4. Ensure that the Kafka and ZooKeeper pods are in the READY state.
5. In the **override-values.yaml** file for **oc-cn-ece-helm-chart**, set the following keys:
   - **charging.server.kafkaEnabledForNotifications**: Set this to **true**.
   - **charging.kafkaConfigurations.kafkaConfigurationList[].name**: Set this to the name of your ECE cluster.
   - **charging.kafkaConfigurations.kafkaConfigurationList[].hostname**: Set this to the host name of the system on which Kafka runs.
   - **charging.kafkaConfigurations.kafkaConfigurationList[].topicName**: Set this to **ECENotifications**.
   - **charging.kafkaConfigurations.kafkaConfigurationList[].suspenseTopicName**: Set this to **ECESuspenseQueue**.
6. From the **helmcharts** directory, install the ECE cloud native service:

   ```
   helm install EceReleaseName oc-cn-ece-helm-chart --namespace BrmNameSpace --values OverrideValuesFile
   ```

The notification topics are created in Apache Kafka.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Creating an Oracle WebLogic Notification Queue

To create notification queues and topics in Oracle WebLogic:

1. Ensure that Oracle WebLogic Server is running in your Kubernetes cluster.
2. Create a separate WebLogic domain for the ECE notification queues.

   > **Note**
   >
   > Do not create the ECE notification queues in an existing WebLogic domain. For example, do not use the Billing Care, Business Operations Center, PDC, or Billing Care REST API domains.

3. Add the following third-party libraries to a customized ECE image:
   - **com.oracle.weblogic.beangen.general.api.jar**
   - **wlthint3client.jar**

   To add the libraries to the image:

   a. Create a **3rdparty_jars** directory in the build context for your customized ECE image.
   b. Copy the required JAR files into the **3rdparty_jars** directory.
   c. In the Dockerfile for the customized ECE image, copy the **3rdparty_jars** directory to **/home/charging/ext/3rdparty_jars**.

      ECE copies JAR files from **/home/charging/ext/3rdparty_jars** into the ECE library directory when the container starts.
4. Build the customized ECE image and push it to a container image repository that your Kubernetes cluster can access.
5. Create an **override-values.yaml** file for **oc-cn-ece-helm-chart**.
6. Configure ECE to use the customized image in the **override-values.yaml** file:
   - **imageRepository**: Set this to the repository server that contains your customized ECE image.
   - **container.image**: Set this to the name and tag of your customized ECE image.
   - **imagePullSecrets**: If the repository requires authentication, set this to the Kubernetes secret that ECE uses to pull the image.
7. If you enable SSL for WebLogic, create a Kubernetes secret that contains the JKS keystore.

   For a disaster recovery environment, provide the common JKS keystore required by the participating site.
8. Set the following keys in your **override-values.yaml** file:
   - **secretEnv.JMSQUEUEPASSWORD**: Set this to the WebLogic user password.
   - **secretEnv.NOTIFYEVENTKEYPASS**: If you enable WebLogic SSL, set this to the KeyStore password.
   - **job.jmsconfig.runjob**: Set this to **true**.
   - **job.jmsconfig.preCreateJmsServerAndModule**: Set this to **true** if the JMS configuration job must create the ECE JMS module and subdeployment.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **charging.server.weblogic.jmsmodule**: Set this to **ECE**.
- **charging.server.weblogic.subdeployment**: Set this to **ECEQueue**.
- **charging.server.kafkaEnabledForNotifications**: Set this to **false**.
- In the applicable **JMSConfiguration** entries, set **HostName**, **Port**, **Protocol**, and **ConnectionURL** to the values for your WebLogic environment.
- If you enable WebLogic SSL:
  - Set **KeyStoreLocation** to the key in the Kubernetes secret that contains the keystore, and
  - Set **extJMSKeyStoreSecret** to the name of the Kubernetes secret.

For more information about these keys, see Table 10-1.

9. From the **helmcharts** directory, install the ECE cloud native service:

   ```
   helm install EceReleaseName oc-cn-ece-helm-chart --namespace BrmNameSpace --values
   OverrideValuesFile
   ```

The ECE JMS configuration creates the following resources in the ECE domain of your WebLogic Server:

- A WebLogic notification topic named **NotificationTopic**.
- A WebLogic notification queue named **SuspenseQueue**
- A WebLogic connection factory named **NotificationFactory**

Next, configure the connection factory so that clients can connect to the ECE notification queues and topics in Oracle WebLogic.

To configure the connection factory:

1. On the WebLogic Server in which the JMS ECE notification queue resides, sign in to WebLogic Remote Console.
2. Click **Edit Tree**, then **Services**, and then **JMS Modules**.

   The Summary of JMS Modules page appears.
3. In the JMS Modules table, click **ECE**.
4. Click **Connection Factories** in the tree in the left pane.
5. In the JMS Connection factories table, click **NotificationFactory**.
6. On the **Client** tab, do the following:
   a. In **Client ID Policy**, select **Unrestricted**.
   b. In **Subscription Sharing Policy**, select **Sharable**.
   c. Click **Save**.
7. On the **Transactions** tab, do the following.
   a. In **Transaction Timeout**, enter **2147483647**, which is the maximum timeout value.
   b. Click **Save**.
8. Click the shopping cart at the top right, and then click **Commit Changes** to commit your changes.

For more information, see Oracle WebLogic Remote Console Online Help.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Configuring ECE for a Multischema BRM Environment

If your BRM database contains multiple schemas, you must configure ECE to connect to each schema.

To configure ECE for a BRM multischema database:

1. Open your **override-values.yaml** file for the **oc-cn-ece-helm-chart** chart.
2. Specify the password for accessing each schema in the BRM database. To do so, configure these keys for each schema:
   - **secretEnv.BRMDATABASEPASSWORD.schema**: Set this to the schema number. Enter **1** for the primary schema, **2** for the secondary schema, and so on.
   - **secretEnv.BRMDATABASEPASSWORD.PASSWORD**: Set this to the schema password.

   This shows example settings for two schemas:

   ```
   secretEnv:
      BRMDATABASEPASSWORD:
         - schema: 1
           PASSWORD: EncryptedPassword
         - schema: 2
           PASSWORD: EncryptedPassword
   ```

3. Configure a customerUpdater pod for each schema. To do so, add a **-schemaNumber** list for each schema. In the list:
   - Set the **SchemaNumber** key to **1** for the primary schema, **2** for the secondary schema, and so on.
   - Set the **amtAckQueueName** key to the fully qualified name of the acknowledgment queue to which the **pin_amt** utility listens to Account Migration Manager (AMM)-related acknowledgment events. The value is in the format *primarySchema*.**ECE_AMT_ACK_QUEUE**, where *primarySchema* is the name of the primary schema.
   - Set the **hostName** and **jdbcUrl** keys to their corresponding values for each schema.

   This shows example settings for two schemas:

   ```
   customerUpdater:
      customerUpdaterList:
         - schemaNumber: "1"
           coherenceMemberName: "customerupdater1"
           replicas: 1
           jmxEnabled: true
           coherencePort: ""
           jvmGCOpts: ""
           jvmJMXOpts: ""
           jvmCoherenceOpts: ""
           jvmOpts: ""
           jmxport: ""
           restartCount: "0"
           oracleQueueConnectionConfiguration:
              name: "customerupdater1"
              gatewayName: "customerupdater1"
              hostName: ""
              port: "1521"
              sid: "pindb"
              userName: "pin"
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
            jdbcUrl: ""
            queueName: "IFW_SYNC_QUEUE"
            suspenseQueueName: "ECE_SUSPENSE_QUEUE"
            ackQueueName: "ECE_ACK_QUEUE"
            amtAckQueueName: "pin0101.ECE_AMT_ACK_QUEUE"
            batchSize: "1"
            dbTimeout: "900"
            retryCount: "10"
            retryInterval: "60"
            walletLocation: "/home/charging/wallet/ecewallet/"

         - schemaNumber: "2"
           coherenceMemberName: "customerupdater2"
           replicas: 1
           jmxEnabled: true
           coherencePort: ""
           jvmGCOpts: ""
           jvmJMXOpts: ""
           jvmCoherenceOpts: ""
           jvmOpts: ""
           jmxport: ""
           oracleQueueConnectionConfiguration:
              name: "customerupdater2"
              gatewayName: "customerupdater2"
              hostName: ""
              port: "1521"
              sid: "pindb"
              userName: "pin"
              jdbcUrl: ""
              queueName: "IFW_SYNC_QUEUE"
              suspenseQueueName: "ECE_SUSPENSE_QUEUE"
              ackQueueName: "ECE_ACK_QUEUE"
              amtAckQueueName: "pin0101.ECE_AMT_ACK_QUEUE"
              batchSize: "1"
              dbTimeout: "900"
              retryCount: "10"
              retryInterval: "60"
              walletLocation: "/home/charging/wallet/ecewallet/"
```

4. Configure a ratedEventFormatter pod for processing rated events belonging to each BRM schema. To do so, add a **-schemaNumber** list for each schema. In the list, set the **schemaNumber** and **partition** keys to **1** for the primary schema, **2** for the secondary schema, and so on.

   This shows example settings for two schemas:

```
ratedEventFormatter:
   ratedEventFormatterList:
      - schemaNumber: "1"
        replicas: 1
        coherenceMemberName: "ratedeventformatter1"
        jmxEnabled: true
        coherencePort:
        jvmGCOpts: ""
        jvmJMXOpts: ""
        jvmCoherenceOpts: ""
        jvmOpts: ""
        jmxport: ""
        restartCount: "0"
        ratedEventFormatterConfiguration:
           name: "ratedeventformatter1"
           primaryInstanceName: "ratedeventformatter1"
           partition: "1"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
            noSQLConnectionName: "noSQLConnection"
            connectionName: "oraclePersistence1"
            threadPoolSize: "6"
            retainDuration: "0"
            ripeDuration: "600"
            checkPointInterval: "6"
            maxPersistenceCatchupTime: "60"
            siteName: ""
            pluginPath: "ece-ratedeventformatter.jar"
            pluginType:
"oracle.communication.brm.charging.ratedevent.formatterplugin.internal.BrmCdrPluginDi
rect"
            pluginName: "brmCdrPlugin1"
            noSQLBatchSize: "25"

      - schemaNumber: "2"
        replicas: 1
        coherenceMemberName: "ratedeventformatter2"
        jmxEnabled: true
        coherencePort:
        jvmGCOpts: ""
        jvmJMXOpts: ""
        jvmCoherenceOpts: ""
        jvmOpts: ""
        jmxport: ""
        ratedEventFormatterConfiguration:
           name: "ratedeventformatter2"
           primaryInstanceName: "ratedeventformatter2"
           partition: "2"
           noSQLConnectionName: "noSQLConnection"
           connectionName: "oraclePersistence1"
           threadPoolSize: "6"
           retainDuration: "0"
           ripeDuration: "600"
           checkPointInterval: "6"
           maxPersistenceCatchupTime: "60"
           siteName: ""
           pluginPath: "ece-ratedeventformatter.jar"
           pluginType:
"oracle.communication.brm.charging.ratedevent.formatterplugin.internal.BrmCdrPluginDi
rect"
           pluginName: "brmCdrPlugin1"
           noSQLBatchSize: "25"
```

5. Save and close your **override-values.yaml** file for **oc-cn-ece-helm-chart**.
6. In your **override-values.yaml** file, configure **charging.itemAssignmentConfig.poidIdConfigurations[]** by specifying **schemaName** and **poidQuantity** for each schema.

   This shows example settings for three schemas:

```
<itemAssignmentConfigconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.item.ItemAssignmentCo
nfig" itemAssignmentEnabled="true" delayToleranceIntervalInDays="0"
poidPersistenceSafeCount="12000">
   <schemaConfigurationGroup config-class="java.util.ArrayList">
      <poidIdConfigurationconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.item.PoidIdConfigurat
ion" schemaName="1" poidQuantity="2000000">
      </poidIdConfiguration>
      <poidIdConfigurationconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.item.PoidIdConfigurat
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
ion" schemaName="2" poidQuantity="2000000">
      </poidIdConfiguration>
      <poidIdConfigurationconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.item.PoidIdConfigurat
ion" schemaName="3" poidQuantity="2000000">
      </poidIdConfiguration>
   </schemaConfigurationGroup>
</itemAssignmentConfig>
```

After you deploy **oc-cn-ece-helm-chart** in "Deploying BRM Cloud Native Services", the ECE pods will be connected to your BRM database schemas.

Configuring ECE for Overload Protection

When the amount of incoming requests for ECE cloud native exceeds its capacity, it experiences an overload. This can lead to charging nodes becoming stuck or unresponsive. To avoid this, you can enable an overload protection mechanism.

When an overload happens, ECE starts rejecting the new incoming requests and informs the client about any submitted request that could not be processed. These requests can then be routed to some other alternative gateway or can be accepted by the same gateway once the overload condition is resolved.

You can configure ECE cloud native either to reject all incoming requests or to reject requests based on priority. The priority level is determined by the overload condition which is calculated based on queue depth and average latency. A Threshold Overload Monitor independently monitors both factors using an exponentially weighted moving average (EWMA) combined with hysteresis. For more information, see "About System Overload Protection" in *BRM System Administrator's Guide*.

> **Note**
>
> Priority-based overload protection is applicable only to HTTP Gateway and Diameter Gateway.

You can configure ECE cloud native to monitor thresholds at different levels based on your expected throughput and latency:

1. Open your Helm Chart's **values.yaml** file.
2. Define the attributes in **overloadConfigurations**.

   For more information about the attributes, see "About the overloadConfigurations Attribute Entries" in *BRM System Administrator's Guide*.

If you want ECE to reject all the incoming requests, enable the overload protection and do not add any other parameters.

Configuring Direct Loading Using the ECE RE Manager Plug-in

By default, the ECE Helm chart configures RE Formatter to use the ECE RE Manager plug-in in DIRECT mode. In this mode, the plug-in loads rated events directly into the BRM database.

Before you deploy ECE, configure the BRM database connection used for direct loading.

To configure the BRM database connection for direct loading:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Open the **override-values.yaml** file for **oc-cn-ece-helm-chart**.
2. Configure **REFPluginConfiguration.jdbcConfigurations** with the JDBC URL or URLs for the BRM database schemas.

   For a single-schema BRM deployment, configure one JDBC connection:

   ```
   REFPluginConfiguration:
     jdbcConfigurations:
       - connectionURL: "jdbc:oracle:thin:@//<host>:<port>/<service>"
         user: "pin"
         wallet_entry_name: "brm.db.pwd"
   ```

   where:

   - <*host*> is the host name of the BRM database.
   - <*port*> is the database listener port.
   - <*service*> is the BRM database service name.

   For a multi-schema BRM deployment, configure one JDBC connection for each BRM schema. Each RE Formatter uses the JDBC connection that points to its corresponding BRM schema:

   ```
   REFPluginConfiguration:
     jdbcConfigurations:
       - connectionURL: "jdbc:oracle:thin:@//<host>:<port>/<service>"
         user: "pin"
         wallet_entry_name: "brm.db.pwd"
       - connectionURL: "jdbc:oracle:thin:@//<host>:<port>/<service>"
         user: "pin"
         wallet_entry_name: "brm.db.pwd"
   ```

   where:

   - <*host*> is the host name of the BRM database.
   - <*port*> is the database listener port.
   - <*service*> is the BRM database service name.
3. Deploy your ECE cloud native services by following the instructions in "Deploying BRM Cloud Native Services".

Table 10-3 lists the **override-values.yaml** properties that configure direct loading from ECE when you use that file. If your deployment does not use **override-values.yaml**, configure the same properties in **values.yaml**. The ECE Helm chart supplies the default values.

Table 10-3 ECE Direct Loading Configuration Properties

Attribute Name	Default Value	Description
charging.remEnabled	**true**	Enables the ECE RE Manager plug-in.
REFPluginConfiguration.refmode	**DIRECT**	Sets the initial rated-event transfer mode to direct database loading.

Copyright © 2019, 2026, Oracle and/or its affiliates.

11

Deploying and Configuring the Mapper Framework

Learn how to deploy and configure the Mapper Framework to tailor Oracle Communications Billing and Revenue Management (BRM) cloud native deployments to your organization's requirements. You can integrate custom mapping logic into your BRM Cloud Native environment, leveraging both Helm and Kubernetes tooling.

Topics in this document:

- Overview of the Mapper Framework
- Deploying Custom Mapper Files
- Configuring the Mapper in BRM Helm Charts
- Mapper Framework Configuration Keys
- Example Usage in override-values.yaml
- Best Practices for Managing Mapper Files

Overview of the Mapper Framework

The BRM Cloud Native Mapper framework enables you to implement custom mapping logic for your deployment. Use mapper files, defined in YAML format, to control how APIs transform data to and from flist structures. This approach lets you customize field mappings, perform type conversions, and apply conditional logic, all without modifying the core application code. You can deploy, update, verify, and roll back mapper files as needed to support your business requirements.

Deploying Custom Mapper Files

Custom mapper files can be deployed at any stage of the BRM Cloud Native life cycle: initial deployment, upgrades, or at runtime for dynamic updates. You can choose a method based on your deployment scenario and file size.

There are two primary approaches:

- For files less than 1 MiB:
  - **Deploying Mapper Files Using Helm**
    To deploy mapper files smaller than 1 MiB, place them in the **oc-cn-helm-chart/rsm/extendedMappers** directory. The files are automatically packaged and loaded when the relevant pods start.
  - **Deploying Mapper Files Using Kubernetes ConfigMaps**
    You can deploy mapper files dynamically or at runtime by creating Kubernetes ConfigMaps. This method is recommended for files that require updating without restarting pods, or for YAML files that change frequently.

    To create a ConfigMap from a single YAML file, run:

    ```
    kubectl create configmap configmap-name --from-file=yaml-file-name -n namespace
    ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

where:

* *configmap-name* is the name assigned to the ConfigMap.
* *yaml-file-name* is the name of the custom mapper YAML file.
* *namespace* is the Kubernetes namespace where the Revenue Stream Manager (RSM) is deployed.

To create a ConfigMap from all YAML files in a directory, run:

```
kubectl create configmap configmap-name --from-file=. -n namespace
```

where **--from-file=.** includes all files in the current directory.

> **Note**
>
> Run all commands from the directory containing the files.

- For files more than 1 MiB:
  - **Deploying Mapper Files Using Custom SDK Images**
    For mapper files larger than 1 MiB, add these files to the **/SDK/extendedMappers** directory in your custom SDK image. Build and push the image to your container registry. Reference the new image in your REST Services Manager deployment configuration.

Configuring the Mapper in BRM Helm Charts

To enable and configure a custom mapper, update the **override-values.yaml** file. For example, set the external ConfigMap and trigger a pod restart:

```
ocrsm:
  rsm:
    externalExtendedMapper: "configmap-name"
    configEnv:
      restartVersion: version-number
```

where:

- **externalExtendedMapper** specifies the name of the ConfigMap that contains your custom mapper files.
- **restartVersion** is a numeric or string value. Increment this value in your deployment and run a Helm upgrade to apply the changes.

> **Note**
>
> - Reference your custom ConfigMap in the **externalExtendedMapper** key if you create the ConfigMap.
> - For SDK image-based extensions, ensure that you correctly define the custom image path in the image configuration.
>
> See "Configuring and Adding Custom Mapper Files" for more information about updating **override-values.yaml** with the path to your custom mapper files.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Mapper Framework Configuration Keys

Configure the Mapper Framework to enable custom field mapping and deployment settings for BRM REST Services Manager (RSM):

```
ocrsm:
   rsm:
      mapperConfiguration:
         config:
            paths:
               event:
                  v5:
                     ProductCreateEvent:
                        mapper-specification-keys:
                           default: Product
                           Event:
                              execute:
                                 - "mapper.brm.v5.tmf637.productCreateEvent.execute"
                              request:
                                 - "mapper.brm.v5.tmf637.productCreateEvent.request"
                              response:
                                 - "mapper.brm.v5.tmf637.productCreateEvent.response"
                                 - "mapper.custom.v5.tmf637.productCreateEvent.response"
                           Product:
                              request:
                                 - "mapper.brm.v5.tmf637.post.request.product"
                                 - "mapper.brm.v5.tmf637.post.request.extendedProduct"
                                 - "mapper.custom.v5.tmf637.post.request.product"
                              response:
                                 - "mapper.brm.v5.tmf637.post.response.product"
                                 - "mapper.brm.v5.tmf637.post.response.extendedProduct"
                                 - "mapper.custom.v5.tmf637.post.response.product"
```

where:

- **ProductCreateEvent** specifies the event type.
- **mapper-specification-keys** defines logical profiles or mapping variants.
- **default** applies when no explicit profile is provided.
- **execute**, **request**, and **response** specify the mapping configuration files for the execution opcode, request transformation, and response transformation, respectively.

Example Usage in override-values.yaml

To load your custom Mapper and Kafka configurations during deployment, add the relevant configuration keys directly to your main **override-values.yaml** file. Place these keys at the root level of the file to align with the primary chart hierarchy.

When updating mapper files, increment the **restartVersion** value and run the **helm upgrade** command to ensure the pods reload the new configuration. Always follow security best practices by protecting sensitive values, such as passwords, and managing access controls appropriately.

The following example demonstrates how to specify these configuration settings in your **override-values.yaml** file:

```
ocrsm:
  rsm:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    isEnabled: true
    externalExtendedMapper: custom-mapper-configmap
    configEnv:
      restartVersion: 2

ocbrm:
  dm_kafka:
    isEnabled: true
    deployment:
      kafka_bootstrap_server_list: ece-kafka:9093
      topicName: BRM
      isSecurityEnabled: true
      trustStorePassword: password
```

Best Practices for Managing Mapper Files

To ensure reliability, traceability, and rapid recovery, follow these best practices when managing mapper files in your environment:

- **Version Control:** Store all custom mapper YAML files in a version control system with clear branching and review policies.
- **Testing:** Validate new or modified mappings in a non-production environment using automated or manual tests before production rollout.
- **CI/CD:** Automate ConfigMap creation, SDK image builds, and deployment of mapper files via CI/CD pipelines.
- **Rollback Planning:** Maintain a rollback plan to restore the last stable configuration promptly if errors occur after deployment.

Copyright © 2019, 2026, Oracle and/or its affiliates.

12

Deploying BRM Cloud Native Services

Learn how to deploy Oracle Communications Billing and Revenue Management (BRM) cloud native services by running the Helm install command.

Topics in this document:

- Deploying BRM Cloud Native Services

Deploying BRM Cloud Native Services

> **Note**
>
> The **oc-cn-init-db-helm-chart** and **oc-cn-helm-chart** charts must be deployed in different namespaces.

To deploy BRM cloud native services, do this:

1. Create a namespace for the BRM Helm chart.

   ```
   kubectl create namespace BrmNameSpace
   ```

   where *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.

2. Validate the content of your Helm charts by using the **Helm lint** command.

   - For Helm 3.6.0 and later releases, enter these commands from the **helmcharts** directory:

     ```
     helm lint --strict oc-cn-helm-chart --values oc-cn-helm-chart/values.yaml --
     values OverrideValuesFile
     helm lint --strict oc-cn-ece-helm-chart --values oc-cn-ece-helm-chart/
     values.yaml --values OverrideValuesFile
     helm lint --strict oc-cn-op-job-helm-chart --values oc-cn-op-job-helm-chart/
     values.yaml --values OverrideValuesFile
     ```

   - For previous Helm releases, enter these commands from the **helmcharts** directory:

     ```
     helm lint --strict oc-cn-helm-chart
     helm lint --strict oc-cn-ece-helm-chart
     helm lint --strict oc-cn-op-job-helm-chart
     ```

   You'll see this if the commands complete successfully:

   ```
   3 chart(s) listed, no failures
   ```

3. If you are using Pricing Design Center (PDC) or Business Operations Center, do this:

   a. Ensure BRM images are available and BRM is deployed successfully, for PDC pods to deploy successfully

   b. Direct WebLogic Kubernetes Operator to monitor the BRM namespace:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
helm upgrade weblogic-operator  weblogic-operator/weblogic-operator \
--namespace Operator \
--reuse-values \
--set "domainNamespaces={BrmNameSpace}" \
--wait
```

where *Operator* is the namespace you created for WebLogic Kubernetes Operator as part of the prerequisite tasks.

c. Create WebLogic domains by entering this command from the **helmcharts** directory:

```
helm install OpJobReleaseName oc-cn-op-job-helm-chart --namespace BrmNameSpace --values OverrideValuesFile
```

where *OpJobReleaseName* is the release name for **oc-cn-op-job-helm-chart** and is used to track this installation instance. It must be different from the one used for the BRM Helm chart.

4. Install BRM cloud native services by entering this command from the **helmcharts** directory:

```
helm install BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
```

where *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance. It must be different from the one used for **oc-cn-init-db-helm-chart**.

5. To install the ECE cloud native service, enter this command from the **helmcharts** directory:

```
helm install EceReleaseName oc-cn-ece-helm-chart --namespace BrmNameSpace --values OverrideValuesFile
```

where *EceReleaseName* is the release name for **oc-cn-ece-helm-chart** and is used to track this installation instance. It must be different from the one used for the BRM Helm chart.

Copyright © 2019, 2026, Oracle and/or its affiliates.

13

Deploying into Oracle Cloud Infrastructure

Learn how to deploy Oracle Communications Billing and Revenue Management (BRM) cloud native services into Oracle Cloud Infrastructure.

Topics in this document:

- Deploying into Oracle Cloud Infrastructure

Deploying into Oracle Cloud Infrastructure

Oracle Cloud Infrastructure is a set of complementary cloud services that enable you to run a wide range of applications and services in a highly available hosted environment. It offers high-performance computing capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premise network. BRM cloud native deployment is tested in Oracle Cloud Infrastructure for the following services both on Virtual Machine and Bare Metal:

- BRM cloud native application and database running on IaaS
- BRM cloud native application managed by Oracle Kubernetes Engine and database on IaaS
- BRM cloud native application managed by Oracle Kubernetes Engine and database on DBaaS

Deploying the BRM cloud native services into Oracle Cloud Infrastructure involves these high-level steps:

> **Note**
>
> These are the bare minimum tasks for deploying BRM cloud native services in Oracle Cloud Infrastructure. Your steps may vary from the ones listed below.

1. Sign up for Oracle Cloud Infrastructure.
2. Create a database system on a bare metal or virtual machine instance.

   Select a database version that is compatible with the BRM cloud native software requirements. See *BRM Compatibility Matrix*.
3. Create a Kubernetes cluster and deselect the **Tiller (Helm) Enabled** option. The version of Helm used by Oracle Cloud Infrastructure isn't compatible with the BRM cloud native software requirements.
4. Install and configure the Oracle Cloud Infrastructure Command Line Interface (CLI).

   CLI is a small footprint tool that you can use on its own or with the Console to complete OCI tasks. It's needed here to download the **kubeconfig** file.
5. Install and configure **kubectl** on your system to perform operations on your cluster in Oracle Cloud Infrastructure.

Copyright © 2019, 2026, Oracle and/or its affiliates.

6. The **kubeconfig** file (by default named **config** and stored in the **$HOME/.kube** directory) provides the necessary details to access the cluster using **kubectl** and the Kubernetes Dashboard.

   Download **kubeconfig** to access your cluster on Oracle Cloud Infrastructure by entering this command:

   ```
   oci ce cluster create-kubeconfig --cluster-id ClusterId --file $HOME/.kube/
   config --region RegionId --token-version 2.0.0
   ```

   where *ClusterId* is the Oracle Cloud Identifier (OCID) of the cluster, and *RegionId* is the region identifier such as us-phoenix-1 and us-ashburn-1.

7. Set the **$KUBECONFIG** environment variable to the downloaded **kubeconfig** file by entering this command:

   ```
   export KUBECONFIG=$HOME/.kube/config
   ```

8. Verify access to your cluster. You can enter this command and then match the output Internal IP Addresses and External IP Addresses against the nodes in your cluster in the Oracle Cloud Infrastructure Console.

   ```
   kubectl get node -o wide
   ```

9. Download and configure Helm in your local system.

10. Place the BRM cloud native Helm chart on your system where you have downloaded and configured **kubectl** and Helm. Then, follow the instructions in "Configuring and Deploying BRM Cloud Native" in *BRM Cloud Native Deployment Guide*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

14

Uninstalling Your BRM Cloud Native Deployment

Learn how to uninstall the Oracle Communications Billing and Revenue Management (BRM) cloud native deployment from your system.

Topics in this document:

- Uninstalling Your BRM Cloud Native Deployment
- Uninstalling Selected BRM Cloud Native Services

Uninstalling Your BRM Cloud Native Deployment

When you uninstall a Helm chart from your BRM cloud native deployment, it removes only the Kubernetes objects that it created during installation.

To uninstall, enter this command:

```
helm delete ReleaseName --namespace NameSpace
```

where:

- *ReleaseName* is the name you assigned to this installation instance.
- *NameSpace* is the namespace in which the BRM Kubernetes objects reside.

Uninstalling Selected BRM Cloud Native Services

Depending on the scenario, you might need to temporarily or permanently uninstall BRM cloud native services, such as Billing Care, the Billing Care REST API, or Business Operations Center, while retaining other BRM services. To do this, you upgrade your **oc-cn-helm-chart** release by disabling the service you intend to remove.

For example, to remove only the Billing Care REST API service, you would set the **ocbc.bcws.isEnabled** key to **false** in your **override-values.yaml** file and then upgrade your release of **oc-cn-helm-chart**:

```
helm upgrade --namespace NameSpace ReleaseName oc-cn-helm-chart --values override-values.yaml
```

This would bring down the WebLogic servers that are hosting the Billing Care REST API and remove all resources created for this service through **oc-cn-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Part IV
Customizing BRM Cloud Native

This part provides information about customizing Oracle Communications Billing and Revenue Management (BRM) cloud native. It contains the following chapters:

- Customizing BRM Cloud Native Services
- Building Your Own Images

Copyright © 2019, 2026, Oracle and/or its affiliates.

15

Customizing BRM Cloud Native Services

Learn how to customize the Oracle Communications Billing and Revenue Management (BRM) server and clients in a cloud native environment to meet your business needs.

The Podman build commands in this chapter reference Dockerfile and related scripts as is from the **oc-cn-docker-files-16.26.0.*x.x*.tgz** package. Ensure you use your own version of Dockerfile and related scripts before running the build command.

Topics in this document:

- Customizing BRM Server
- Customizing Billing Care
- Customizing ECE

> ⚠ **Caution**
>
> The Dockerfiles and related scripts are provided for reference only. You can refer to them to build or extend your own images. Support is restricted to core product issues only and no support will be provided for custom Dockerfiles and scripts.

Customizing BRM Server

You can customize BRM Server by layering the BRM cloud native image with a customized library file.

For example, you could extend the **fm_subscription_pol_custom.so** library file and layer it with the BRM cloud native image by doing this:

1. Customize your **lib/fm_subscription_pol_custom.so** library file as follows:
   a. Enable the BRM SDK by setting the following keys in your **override-values.yaml** file for **oc-cn-helm-chart**:

```
brm_sdk:
   isEnabled: true
   deployment:
      imageName: brm_sdk
      imageTag: 16.26.0.x.x
   volume:
      storage: 50Mi
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> Pip is no longer installed by default in BRM SDK images. If you require pip, you can install it by building a derived image using the following sample Dockerfile:
>
> ```
> FROM brm_sdk:16.26.0.x.x
> USER root
> RUN /usr/local/bin/python3 -m ensurepip --upgrade --default-pip && \
>     /usr/local/bin/python3 -m pip --version
> USER omsuser
> ```

b. By default, the brm-sdk pod uses dynamic volume provisioning. To use a static volume instead, add the **createOption** keys under **brm_sdk.volume**. For example:

```
brm_sdk:
  volume:
    storage: 50Mi
    createOption:
      hostpath:
        path: pathOnNode
        type: Directory
```

where *pathOnNode* is the location on the host system of the external PV.

c. Run the **helm upgrade** command to deploy the brm-sdk pod:

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values
OverrideValuesFile --namespace BrmNameSpace
```

d. Run the following **kubectl** command to retrieve the brm-sdk pod name:

```
kubectl get pods --namespace BrmNameSpace | grep brm-sdk
```

If successful, you should see something similar to this:

```
NAME                        READY   STATUS    RESTARTS   AGE
brm-sdk-f67b95777-bf8j5     1/1     Running   0          18m
```

e. Run the following **kubectl** command to retrieve the name of the PVC volume for brm-sdk:

```
kubectl get pvc --namespace BrmNameSpace | grep brm-sdk
```

If successful, you should see something similar to this:

```
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
brm-sdk   Bound    pvc-094feae0-4d11-4887-83a0-b47a0fc6a3f4   50Mi
RWX            myclass        23h
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

f. List the files and folders in **/mnt/oke_test/brm** to verify that the PVC volume is present:

```
ls /mnt/oke_test/brm/ | grep pvc-094feae0-4d11-4887-83a0-b47a0fc6a3f4
```

If successful, you should see something similar to this:

```
brm-sdk-pvc-094feae0-4d11-4887-83a0-b47a0fc6a3f4
```

g. Do one of the following:

- Copy the custom C file to the PVC:

  cp *customFile nfsMountPath/BrmNameSpace/pvcVolumePath/*

  For example:

  ```
  cp fm_cust_pol_valid_billinfo.c /mnt/oke_test/brm/brm-sdk-
  pvc-094feae0-4d11-4887-83a0-b47a0fc6a3f4/
  ```

- Copy the custom C file to the **oc-cn-helm-chart/brm_sdk_scripts/** directory and run the **helm upgrade** command:

  cp *customFile* **oc-cn-helm-chart/brm_sdk_scripts/**
  helm upgrade *BrmReleaseName* **oc-cn-helm-chart --namespace** *BrmNameSpace* **--values oc-cn-helm-chart/override_values.yaml**

  For example:

  ```
  cp fm_cust_pol_valid_billinfo.c  oc-cn-helm-chart/brm_sdk_scripts/
  helm upgrade release oc-cn-helm-chart --namespace brm --values oc-
  cn-helm-chart/override_values.yaml
  ```

  The files from **oc-cn-helm-chart/brm_sdk_scripts/** will be present at **/oms/load** in the brm-sdk pod.

h. Run the **kubectl** command to get a shell to a running container:

kubectl exec --namespace *BrmNameSpace* **-it** *brmSDKPod* **bash**

For example:

```
kubectl exec --namespace brm -it brm-sdk-f67b95777-bf8j5 bash
```

i. Build your custom library file in one of these ways:

- If you copied your custom C file to the PVC in step 1.g, do this:

  cd source/sys/*libraryName*
  cp /oms/ext/*fileName* **.**
  make

Copyright © 2019, 2026, Oracle and/or its affiliates.

For example:

```
cd source/sys/fm_cust_pol/
cp /oms/ext/fm_cust_pol_valid_billinfo.c .
make
```

- If you copied your custom C file to **oc-cn-helm-chart/brm_sdk_scripts/** in step 1.g, do this:

  cd source/sys/*libraryName*
  cp /oms/load/*fileName* **.**
  make

  For example:

  ```
  cd source/sys/fm_cust_pol/
  cp /oms/load/fm_cust_pol_valid_billinfo.c .
  make
  ```

j. Copy the generated library file to the PVC:

   cp *customLibrary*.**so /oms/ext/**

   For example:

   ```
   cp fm_cust_pol_custom.so /oms/ext/
   ```

k. Copy the library file from the PVC to the **$PIN_HOME/lib** directory:

   cp *nfsMountPath/BrmNameSpace/brmSDKPod/customLibrary*.**so $PIN_HOME/lib**

   For example:

   ```
   cp /mnt/oke_test/brm/brm-sdk-pvc-094feae0-4d11-4887-83a0-b47a0fc6a3f4/
   fm_cust_pol_custom.so $PIN_HOME/lib
   ```

2. Build the custom Connection Manager (CM) image using the Dockerfile:

   ```
   FROM cm:16.26.0.x.x
   USER root
   COPY lib/fm_subscription_pol_custom.so $PIN_HOME/lib/fm_subscription_pol_custom.so
   RUN chown -R omsuser:root $PIN_HOME/lib/fm_subscription_pol_custom.so && \
    chmod 755 ${PIN_HOME}/lib/fm_subscription_pol_custom.so
   USER omsuser
   ```

3. Build the BRM Server image by entering this command:

   podman build --format docker --tag cm:*imageTag* **--file Dockerfile_cm .**

4. Push the image to the image repository:

   podman tag cm:*imageTag imageRepository***/cm:***imageTag*
   podman push *imageRepository***/cm:***imageTag*

Copyright © 2019, 2026, Oracle and/or its affiliates.

5. Update the custom image name in the **override-values.yaml** file. For example:

```
cm:
   isEnabled: true
   deployment:
      replicaCount: 1
      imageName: cm
      imageTag: imageTag
```

   where *imageTag* must match the value used in step 4.

6. Add the custom configuration for the CM **pin.conf** to the **configmap_pin_conf_cm.yaml** file.

   For example, for the **fm_cust_pol_custom** library:

```
- cm fm_module ${PIN_HOME}/lib/fm_cust_pol_custom${LIBRARYEXTENSION}
fm_cust_pol_custom_config fm_cust_pol_init pin
```

7. Run the **helm upgrade** command to update the release with the new CM image:

```
helm upgrade BrmReleaseName oc-cn-helm-chart --namespace BrmNameSpace --
values oc-cn-helm-chart/override_values.yaml
```

Customizing Billing Care

Extensibility is one of the biggest features of on-premise Billing Care, and this same extensibility is available in the Billing Care cloud native deployment. You can override the existing Billing Care behavior, such as changing labels and icons, add new flows and screens, and so on, by using the Billing Care SDK.

To use the Billing Care SDK in a cloud native environment, do this:

1. Build the Billing Care SDK WAR the same way as described in "Packaging and Deploying Customizations" in *Billing Care SDK Guide*.
2. Create a Billing Care SDK image by using the Linux image as a base.
3. Update the **override-values.yaml** file to direct **oc-cn-op-job-helm-chart** to deploy the SDK WAR file and link it with Billing Care or the Billing Care REST API WAR after deploying them.

The cloud native package includes all of the scripts necessary to prepare and run an SDK image. For example, if your SDK WAR is named **BillingCareCustomizations.war**, you would build the Billing Care SDK image like this:

1. Go to the **oc-cn-docker-files/ocbc/billing_care_sdk** directory.
2. Copy the **BillingCareCustomizations.war** file to your current working directory (**oc-cn-docker-files/ocbc/billing_care_sdk**).
3. Build the SDK image by entering this command:

```
podman build --format docker --build-arg SDK_WAR=BillingCareCustomizations.war --tag
oracle/billingcare_sdk:16.26.0.x.x  .
```

4. In your **override-values.yaml** file for **oc-cn-op-job-helm-chart**, edit the keys shown in Table 15-1. This directs **oc-cn-op-job-helm-chart** to deploy the Billing Care SDK image

Copyright © 2019, 2026, Oracle and/or its affiliates.

rather than the Billing Care image and to include additional files that are needed for successful deployment of SDK.

Table 15-1 Billing Care SDK Keys

Key	Path	Description
imageName	**ocbc.bc.deployment.sdk** **ocbc.bcws.deployment.sdk**	Set this to **oracle/billingcare_sdk**. This is the name of the image, which must be used for the billingcare pod.
imageTag	**ocbc.bc.deployment.sdk** **ocbc.bcws.deployment.sdk**	Set this to **16.26.0.0.0**. This tags the image used for the billingcare pod.
isEnabled	**ocbc.bc.sdk** **ocbc.bcws.sdk**	Set this key to **true** if you want to deploy SDK.
deployName	**ocbc.bc.sdk** **ocbc.bcws.sdk**	The name of the SDK Library in the **Manifest.MF** file. The default is **BillingCareCustomizations**.

5. Install **oc-cn-op-job-helm-chart** followed by **oc-cn-helm-chart** to customize Billing Care or the Billing Care REST API with SDK.

Customizing ECE

You can customize the ECE image by layering the native image with the customized code.

For example:

```
 > cat Dockerfile_custom_ece
FROM oc-cn-ece:16.26.0.x.x
USER root
#commands that need to be run
USER eceuser
```

To build the image, run this Podman command:

```
podman build --format docker --tag customECE:16.26.0.x.x --file customECEDockerfile .
```

where *customECE* is the name of your custom ECE Helm chart, and *customECEDockerfile* is the name of your custom Dockerfile.

For the Helm chart to take the new custom image for installation, set these keys in your **override-values.yaml** file for the ECE Helm chart:

```
imageRepository: "imageRepo:imagePort"
container:
   image: "customECEImageName"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

16

Building Your Own Images

Learn how to build your own images of the Oracle Communications Billing and Revenue Management (BRM), Elastic Charging Engine (ECE), Pipeline Configuration Center, Pricing Design Center (PDC), Billing Care, and Business Operations Center applications.

The Podman build commands in this chapter reference Dockerfile and related scripts as is from the **oc-cn-docker-files-16.26.0.*x*.*x*.tgz** package. Ensure you use your own version of Dockerfile and related scripts before running the build command.

Topics in this document:

- Building BRM Server Images
- Building BRM REST Services Manager Images
- Building PDC REST Services Manager Images
- Building PDC Images
- Building Pipeline Configuration Center Images
- Building Billing Care Images
- Building Business Operations Center Images
- Building Collections Configuration Center Images

Sample Dockerfiles included in the BRM cloud native deployment package (**oc-cn-docker-files-16.26.0.*x*.*x*.tgz**) are examples that depict how default images are built for BRM. If you want to build your own images, refer to the sample Dockerfiles shipped with the product as a reference. Create your own Dockerfiles and then build your images.

> ⚠ **Caution**
>
> The Dockerfiles and related scripts are provided for reference only. You can refer to them to build or extend your own images. Support is restricted to core product issues only and no support will be provided for custom Dockerfiles and scripts.

Building BRM Server Images

To build images for BRM Server, your staging area ($PIN_HOME) must be available from where the images are built. After you unpack **oc-cn-docker-files-16.26.0.*x*.*x*.tgz**, the BRM Server directory structure will be **oc-cn-docker-files/ocbrm**.

> ⓘ **Note**
>
> If you are using Podman to build your images, pass the **--format docker** flag with the **podman build** command.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Building your own BRM Server images involves these high-level steps:

1. You build the BRM Server base image. See "Building Your BRM Server Base Image".
2. You build images for each BRM Server component. See "Building Images of BRM Server Components".
3. You build the Web Services Manager image. See "Building Web Services Manager Images".
4. You build the BRM REST Services Manager image. See "Building BRM REST Services Manager Images".
5. You containerize the Email Data Manager. See "Containerization of Email Data Manager".
6. You containerize the roaming pipeline. See "Containerization of Roaming Pipeline".
7. You build and deploy Vertex Manager. See "Building and Deploying Vertex Manager".

Building Your BRM Server Base Image

To make your directory structure ready for building base images:

1. Edit the **$PIN_HOME/bin/orapki** binary to replace the staging Java path with ${JAVA_HOME}.
2. Create the **$PIN_HOME/installer** directory.
3. If you're behind a proxy server, set the **$PROXY** variable:

   ```
   export PROXY=ProxyHost:Port
   ```

4. Download the Java binary and then copy it to **$PIN_HOME**. See "BRM Software Compatibility" for the latest supported version of Java.
5. Download the Perl binary and then copy it to **$PIN_HOME**. See "BRM Software Compatibility" for the latest supported version of Perl.
6. For your database client:
   a. Copy **oracle_client_response_file.rsp** (64 bit), **downloadOracleClient.sh**, and **waitForOracleClientInst.sh** from **oc-cn-docker-files/ocbrm/base_images** to **$PIN_HOME**.
   b. Modify these parameters in the **downloadOracleClient.sh** file:
      - ORACLE_CLIENT_ZIP: Enter the binary name.
      - REPOSITORY_URL: Enter the location to fetch the database client binary.
   c. If the db_client binary is already downloaded, copy the binary to the **$PIN_HOME/installer** directory.

After preparing your directory structure, build your BRM Server base image:

- For database client 12CR2 (64 Bit) + Java + Perl, enter this command:

  ```
  podman build --format docker --build-arg PROXY=$PROXY --tag
  db_client_and_java_perl:16.26.0.x.x --file DockerFileLocation/
  Dockerfile_db_client_and_java_perl .
  ```

- For database client 12CR2 (64 Bit) + Java, enter this command:

  ```
  podman build --format docker --build-arg PROXY=$PROXY --tag
  db_client_and_java:16.26.0.x.x --file DockerFileLocation/
  Dockerfile_db_client_and_java .
  ```

- For Java, enter this command:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
podman build --format docker --build-arg PROXY=$PROXY --tag java:16.26.0.x.x --file
DockerFileLocation/Dockerfile_java .
```

- For Java + Perl, enter this command:

```
podman build --format docker --build-arg PROXY=$PROXY --tag java_perl:16.26.0.x.x --
file DockerFileLocation/Dockerfile_java_perl .
```

> **Note**
>
> If you use an existing BRM database with custom-built images, set **ocbrm.use_oracle_brm_images** to **false**.
>
> Do not include populated wallets, private keys, sample CA certificates, or default credentials in your custom images or image-build context. Configure TLS wallet material when you deploy BRM. See "Managing Wallet and KeyStore Certificates" for more information.

Building Images of BRM Server Components

The **oc-cn-docker-files-16.26.0.*x*.*x*.tgz** package includes references to all of the Dockerfiles and scripts needed to build images of BRM Server components (except for **oraclelinux:8**).

To build an image of a BRM Server component:

1. Copy these scripts from the **oc-cn-docker-files/ocbrm** directory to your staging area at **$PIN_HOME**:
   - **entrypoint.sh**
   - **cm/preStopHook.sh_cm**
   - **cm/postStartHook.sh**
   - **cm/updatePassword.sh**
   - **eai_js/preStopHook.sh_eai**
2. Do one of the following:
   - For the batch pipeline, roaming pipeline, and real-time pipeline, copy **entrypoint.sh** to **$PIN_HOME/..**, and copy **$PIN_HOME/../setup/BRMActions.jar** to the **$PIN_HOME/jars** directory.
   - For all other components, copy the **$PIN_HOME/../setup/BRMActions.jar** file to **$PIN_HOME**.

   > **Note**
   >
   > Do not include populated wallets, private keys, sample CA certificates, or default credentials in your custom images or image-build context. Provide TLS wallet material separately when you deploy BRM.

3. Set these environment variables:
   - **$PIN_HOME**: Set this to your staging area.
   - **$PERL_HOME**: Set this to the path to Perl. See "BRM Software Compatibility" for the latest supported version of Perl.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **$JAVA_HOME**: Set this to the Java path. See "BRM Software Compatibility" for the latest supported version of Java.

4. Build the image for your BRM component.

   For example, to build a CM image, enter:

   ```
   podman build --format docker --tag cm:16.26.0.x.x --build-arg
   STAGE_PIN_HOME=$PIN_HOME --build-arg STAGE_JAVA_HOME=$JAVA_HOME --build-arg
   STAGE_PERL_HOME=$PERL_HOME --file DockerfileLocation/Dockerfile .
   ```

   To build a roaming pipeline image, enter:

   ```
   podman build --format docker --tag roam_pipeline:$BRM_VERSION --build-arg
   STAGE_PERL_HOME=StagePerlPath .
   ```

   where *StagePerlPath* is the path to the Perl files in your staging area at $PIN_HOME.

   To build a dm-oracle image, enter:

   ```
   podman build --format docker --force-rm=true --no-cache=true --tag
   dm_oracle:16.26.0.x.x --file DockerfileLocation/Dockerfile .
   ```

   where *DockerfileLocation* is the path to the Dockerfiles for your BRM component.

   > **Note**
   >
   > Build batch and realtime pipeline images from the **$PIN_HOME/..** directories.

Building Web Services Manager Images

To containerize images for Web Services Manager, your staging area ($PIN_HOME) must be available from where the Docker images are built.

To build and deploy the Web Services Manager image:

1. Copy these files from the **oc-cn-docker-files** directory to your staging area at **$PIN_HOME**:
   - **brm_wsm_entrypoint.sh**
   - **Dockerfile**
   - **mkstore** (**oc-cn-docker-files/base_images**)
2. Build the Web Services Manager image by entering this command:

   ```
   podman build --format docker --tag brm_wsm:$BRM_VERSION .
   ```

3. Configure your Web services by updating the **configmap_brm_wsm_props.yaml** file.
4. In your **override-values.yaml** file for **oc-cn-helm-chart**, set the **ocbrm.wsm.soap.isEnabled** key to **true**.
5. Deploy the BRM Helm chart:

   ```
   helm install ReleaseName oc-cn-helm-chart --namespace NameSpace --values
   OverrideValuesFile
   ```

   where:

   - *ReleaseName* is the release name, which is used to track this installation instance.
   - *NameSpace* is the namespace in which to create BRM Kubernetes objects.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- *OverrideValuesFile* is the path to the YAML file that overrides the default configurations in the BRM Helm chart's **values.yaml** file.

Containerization of Email Data Manager

The Email Data Manager (DM) enables you to send customer notifications and invoices to your customers through email automatically. The Email DM uses the Sendmail client to forward emails to Postfix, which is the SMTP server. In-turn, Postfix sends the emails to your customers.

The Email DM will have the Sendmail client, and the Kubernetes host will have Postfix running. You must install and configure Postfix on your Kubernetes host.

To configure your cm pod to point to the Email DM, add this key to the **oc-cn-helm-chart/values.yaml** file:

```
ocbrm.dm_email.deployment.smtp: EmailHostName
```

where *EmailHostName* is the hostname of the server on which the Email DM is deployed. For example: em389.us.example.com.

To configure the Kubernetes host or SMTP server to accept data from the Email DM, do this:

1. Log in as the root user to the Kubernetes host.
2. Add the IP address for the Kubernetes host to the **/etc/postfix/main.cf** file:

   ```
   inet_interfaces=localhost, HostIPAddress
   ```

   For example, if the Kubernetes host is 10.242.155.149.

   ```
   inet_interfaces=localhost, 10.242.155.149
   ```

3. Retrieve the container network configuration by running this command on the Kubernetes host:

   ```
   /sbin/ifconfig cni0 | grep netmask | awk '{print$2"\n"$4}'
   ```

   The output will be similar to this:

   ```
   10.244.0.1 ← The Kubernetes host IP, which is in the container network.
   255.255.255.0
   ```

4. Edit the **mynetworks** field in the **/etc/postfix/main.cf** file to include the Kubernetes network in the list of trusted SMTP clients. If the Kubernetes host IP and Email DM container IP are in different networks, add both networks to the **mynetworks** field:

   ```
   mynetworks = TrustedNetworks
   ```

   where *TrustedNetworks* is the IP addresses for the SMTP clients that are allowed to relay mail through Postfix.

   For example:

   ```
   mynetworks = 168.100.189.0/28, 127.0.0.0/8, 10.244.0.0/24
   ```

5. Do one of these:
   - If Postfix is already running in the host, run this command:

     ```
     systemctl restart postfix.service
     ```

   - If Postfix isn't running in the host, run this command:

     ```
     systemctl start postfix.service
     ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> In case of a multi-node environment, you can configure Postfix on the primary node (or any one node).

Containerization of Roaming Pipeline

> **Note**
>
> Do not run multiple replicas of the roampipe pod against the same file system. It is a single-replica pod. To provide high availability with multiple roampipe pods, deploy the pods separately without sharing the same file system, and route incoming CDR files so that each pod processes a separate set of files.

Roaming allows a wireless network operator to provide services to mobile customers from another wireless network. For example, when a mobile customer makes a phone call from outside the home network, roaming allows the customer to access the same wireless services that he has with his home network provider through a visited wireless network operator.

You feed the input files for the roaming pipeline through a Kubernetes PersistentVolumeClaim (PVC). The EDR output files will be available in a PVC for consumption of the rel-daemon pod. When building the roaming pipeline image, pass the Perl path in these files as part of **build-arg**.

To containerize the roaming pipeline, update the **configmap_infranet_properties_rel_daemon.yaml** file to specify how to load your rated CDR output files. For example:

```
batch.random.events TEL, ROAM
ROAM.max.at.highload.time 4
ROAM.max.at.lowload.time  2
ROAM.file.location /oms/ifw/data/roamout
ROAM.file.pattern test*.out
ROAM.file.type STANDARD
```

> **Note**
>
> The input file to the splitter pipeline must start with **Roam_**.

Building and Deploying Vertex Manager

To deploy Vertex Manager (dm-vertex), you layer the dm-vertex image with the libraries for Vertex Communications Tax Q Series (CTQ) or Vertex Sales Tax Q Series (STQ). For the list of supported library versions, see *BRM Compatibility Matrix*.

Deploying with Vertex Communications Tax Q Series

You deploy Vertex Manager with Vertex CTQ by doing the following:

1. Building the new Vertex Manager image by layering it with Vertex CTQ libraries.

Copyright © 2019, 2026, Oracle and/or its affiliates.

a. Copy the entire Vertex CTQ installation directory to the **$PIN_HOME** directory, where **$PIN_HOME** is set to the path of your staging area.

b. Update the paths in the **64bit/bin/ctqcfg.xml**, **64bit/cfg/ctqcfg.xml**, and other Vertex CTQ files present in the Vertex CTQ installation directory. For example:

```
<configuration name="CTQ Test">
<fileControl>
  <updatePath>/oms/vertex/64bit/dat</updatePath>
  <archivePath>/oms/vertex/64bit/dat</archivePath>
  <callFilePath>/oms/vertex/64bit/dat</callFilePath>
  <reportPath>/oms/vertex/64bit/rpt</reportPath>
  <logPath>/oms/vertex/64bit/log</logPath>
</fileControl>
```

c. In your copied Vertex CTQ installation directory, update the **64bit/bin/odbc/odbc.ini** file. For example:

> **Note**
>
> Set the **Driver** and **TNSNamesFile** entries to the file system path inside the pod.

```
[CtqTestOracle]
Description=Vertex, Inc. 8.0 Oracle Wire Protocol
Driver=/oms/vertex/64bit/bin/odbc/lib/VXor827.so
…
HostName=DBhostname
LogonID=DBuser
PortNumber=1521
Password=DBpassword
ServerName=//IPaddress:1521/DBalias
SID=DBalias
TNSNamesFile=/oms/ora_k8/tnsnames.ora
```

where:

- *DBhostname* is the host name of the machine on which the Vertex tax calculation database is installed.
- *DBuser* is the Vertex database schema user name.
- *DBpassword* is the password for the Vertex database schema user.
- *IPaddress* is the IP address of the machine on which the Vertex tax calculation database is installed.
- *DBalias* is the Vertex database alias name, which is defined in your **tnsnames.ora** file.

d. Layer the default images provided by Oracle.

For example, to layer dm-vertex with Vertex CTQ, you could add these sample commands to its Dockerfile. In this example, **$PIN_HOME** is set to **/oms** inside the pod.

```
FROM dm_vertex:16.26.0.x.x

USER root
RUN mkdir -p /oms/vertex/64bit/cfg
RUN chown -R omsuser:root /oms/vertex/64bit/cfg
COPY ./Vertex_CTQ_30206/ /oms/vertex
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
COPY Vertex_CTQ_30206/64bit/lib/libctq.so /oms/lib/
COPY Vertex_CTQ_30206/64bit/bin/odbc/lib/libodbc.so /oms/lib/libodbc.so

RUN chown -R omsuser:root /oms/vertex
RUN chown -R omsuser:root /oms/lib/libctq.so
RUN chown -R omsuser:root /oms/lib/libodbc.so
USER omsuser
```

e. Build your new Vertex Manager image. For example:

```
podman build --format docker --tag dm_vertex_ctq:16.26.0.x.x --file
Dockerfile_vertex_ctq .
```

2. Enabling and configuring Vertex Manager in your BRM cloud native deployment.

a. Set these environment variables in your **oc-cn-helm-chart/templates/dm_vertex.yaml** file:

```
- name: LD_LIBRARY_PATH
  value: "/oms/vertex/64bit/bin/odbc:/oms/lib:/oms/sys/dm_vertex:/oms/vertex/
64bit/lib"
- name: CTQ_CFG_HOME
  value: "/oms/vertex/64bit/bin"
- name: ODBCINI
  value: "/oms/vertex/64bit/bin/odbc/odbc.ini"
```

b. Uncomment these entries in your **oc-cn-helm-chart/templates/configmap_pin_conf_dm_vertex.yaml** file:

```
- dm_vertex commtax_sm_obj ${DM_VERTEX_CTQ_SM}
- dm_vertex commtax_config_name ${DM_VERTEX_CTQ_CFG_NAME}
- dm_vertex commtax_config_path ${DM_VERTEX_CTQ_CFG_PATH}
```

c. Update these key in your **override-values.yaml** file for **oc-cn-helm-chart**:

```
dm_vertex:
  isEnabled: true
  deployment:
    replicaCount: 1
    imageName: dm_vertex_ctq
    imageTag: 16.26.0.x.x
    quantum_db_password: Base64EncodedPassword
    ctqCfg: /oms/vertex/64bit/cfg
    ctqCfgName: CTQ Test
    ctqSmObj: ./dm_vertex_ctq30206.so
```

d. Run the **helm upgrade** command to update your BRM Helm release:

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --
namespace BrmNameSpace
```

where:

- *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.
- *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.
- *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-helm-chart**.

Deploying with Vertex Sales Tax Q Series

You deploy Vertex Manager with Vertex STQ by doing the following:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Copying the required libraries from the Vertex STQ installation directory to your **$PIN_HOME/req_libs** directory.
2. Layer the default images provided by Oracle. For example, to layer dm-vertex with Vertex STQ, you could add these sample commands to its Dockerfile:

```
FROM dm_vertex:16.26.0.x.x

USER root
COPY ["req_libs/libvst*.so", "req_libs/libqutil*.so", "req_libs/libloc*.so",
"/oms/lib/"]
RUN chown omsuser:root -R /oms/lib/ /lib64
USER omsuser
```

3. Build your new Vertex Manager image. For example:

```
podman build --format docker --tag dm_vertex_stq:16.26.0.x.x --file
Dockerfile_vertex_stq .
```

4. Update these key in your **override-values.yaml** file for **oc-cn-helm-chart**:

```
dm_vertex:
  isEnabled: true
  deployment:
    replicaCount: 1
    imageName: dm_vertex
    imageTag: 16.26.0.x.x
    quantum_db_password: Base64EncodedPassword
```

5. Update these entries in your **oc-cn-helm-chart/templates/configmap_env_dm_vertex.yaml** file:

```
SERVICE_FQDN: dm-vertex
QUANTUM_DB_SOURCE: quantum
QUANTUM_DB_SERVER: qsu122a
QUANTUM_DB_USER: quantum
```

6. Update these entries in your **oc-cn-helm-chart/templates/configmap_odbc_ini_dm_vertex.yaml** file:

```
data:
  odbc.ini: |
    [ODBC Data Sources]
    Server = Oracle Server v12.2
    [Server]
    Description = Oracle Server v12.2
    Driver = /usr/lib/oracle/19.20/client64/lib/libsqora.so.19.1
    Servername = PINDB
    UserID = DBuser
    Password = DBpassword
    Port = 1521
    Trace = yes
    TraceFile = /oms_logs/odbc.log
    Database = //DBhostname:DBport
```

where:

- *Server* is the name of the server on which the Vertex database is installed.
- *DBuser* is the Vertex database schema user name.
- *DBpassword* is the password for the Vertex database schema user.
- *DBhostname* is the host name of the machine on which the Vertex tax calculation database is installed.
- *DBport* is the port number of the Vertex tax calculation database.

Copyright © 2019, 2026, Oracle and/or its affiliates.

7. Set these entries in your **oc-cn-helm-chart/templates/configmap_pin_conf_dm_vertex.yaml** file:

```
- dm_vertex quantum_sm_obj ./dm_vertex_stq111.so
- dm_vertex quantumdb_source ${QUANTUM_DB_SOURCE}
- dm_vertex quantumdb_server ${QUANTUM_DB_SERVER}
- dm_vertex quantumdb_user ${QUANTUM_DB_USER}
```

8. Run the **helm upgrade** command to update the BRM Helm release:

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace BrmNameSpace
```

where:

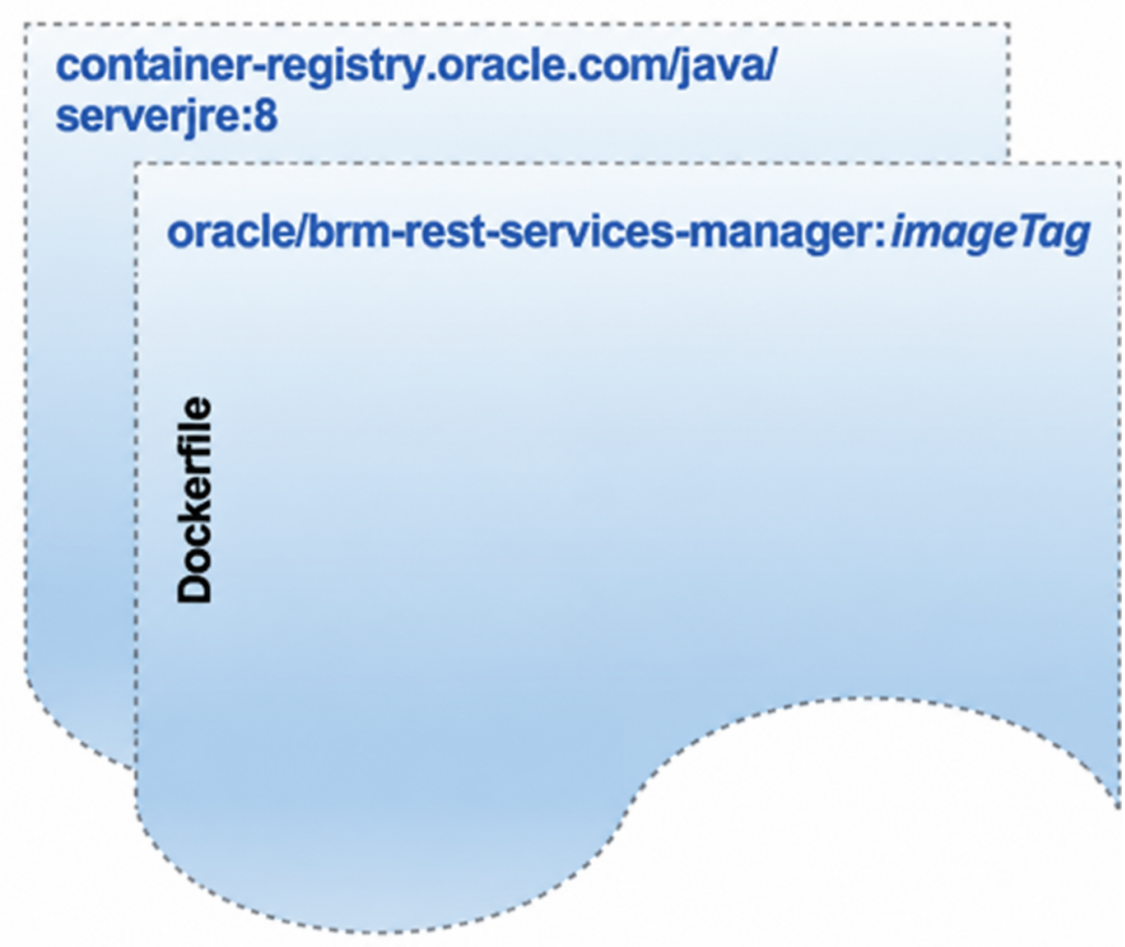
- *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.
- *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.
- *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **values.yaml** file for **oc-cn-helm-chart**.

Building BRM REST Services Manager Images

In a production deployment, containers for BRM REST Services Manager will run in their own pods on a Kubernetes node. Figure 16-1 shows how to stack images for BRM REST Services Manager.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 16-1 Image Stack for BRM REST Services Manager



In this figure:

- **container-registry.oracle.com/java/serverjre:8**: The base image on which BRM REST Services Manager will be deployed. The official image is available at https://container-registry.oracle.com/.
- **oracle/brm-rest-services-manager:16.26.0.*x*.*x***: The sample Dockerfile and related scripts used for creating the BRM REST Services Manager image (**oracle/brm-rest-services-manager:16.26.0.*x*.*x***).

The **oc-cn-docker-files/ocrsm/brm_rest_services_manager** directory in the **oc-cn-docker-files-16.26.0.*x*.*x*.tgz** package contains a Dockerfile, container scripts, and an API JAR file.

You can load or build the BRM REST Services Manager image in the following ways:

- The **oc-cn-brm-rest-services-manager-16.26.0.*x*.*x*.tar** image is included in the package. Apply the image in your machine by running this command:

  ```
  podman load < oc-cn-brm-rest-services-manager-16.26.0.x.x.tar
  ```

- If the image needs customization, modify the Dockerfile and then deploy it using this command:

  ```
  podman build --format docker --tag oracle/brm-rest-services-manager:16.26.0.x.x .
  ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Building PDC REST Services Manager Images

In a production deployment, containers for PDC REST Services Manager will run in their own pods on a Kubernetes node. You create PDC REST Services Manager images by stacking these Dockerfiles in the following order:

1. **container-registry.oracle.com/java/serverjre:8**: The base image on which PDC REST Services Manager will be deployed. The official image is available at https://container-registry.oracle.com/.
2. **oracle/pdcrsm:16.26.0.*x*.*x*.*x***: The sample Dockerfile and related scripts used for creating the PDC REST Services Manager image (**oracle/pdcrsm:16.26.0.*x*.*x*.*x***).

To build PDC REST Services Manager images:

1. Copy the Dockerfile and the **oc-cn-pdc-rsm-jars-16.26.0.*x*.*x*.*x*** file into the current working directory.
2. Run the following commands:

```
tar xvf oc-cn-pdc-rsm-jars-16.26.0.x.x.x.tar
podman build --format docker --tag oracle/pdcrsm:16.26.0.x.x.x .
```

Building PDC Images

To build the PDC image:

1. Download **PricingDesignCenter-16.26.0.*x*.*x*.*x*.zip** to the *ParentFolder*/**Docker_files/PDCImage/other-files** directory.
2. Pull the Java Image from the Oracle Container Registry (https://container-registry.oracle.com). This image is regularly updated with the latest security fixes. You can pull this image to your local system, where you will build other images, with the name **container-registry.oracle.com/java/serverjre:***JavaVersion*.

   where *JavaVersion* is the Oracle Java version number. See *BRM Compatibility Matrix* for supported versions.
3. Set the following environment variables:
   - HTTP_PROXY: Set this to the host name or IP address of your proxy server
   - JAVA_VERSION: Set this to **container-registry.oracle.com/java/serverjre:***JavaVersion*
   - BRM_VERSION: Set this to **16.26.0.*x*.*x*.*x***
4. Build your Oracle PDC BRM integration image by entering this command from the *ParentFolder*/**Docker_files/PDCImage** directory:

```
podman build --format docker --force-rm=true --no-cache=true --build-arg
HTTP_PROXY=$HTTP_PROXY --build-arg JAVA_VERSION=$JAVA_VERSION --tag $IMAGE_NAME --
file Dockerfile .
```

5. (Optional) To use custom fields in your PDC RUM expressions:
   a. Create a **custom_flds.h** file that contains your custom fields.

      For information about the syntax to use in a header file, view the *BRM_home*/**include/pin_flds.h** file in the brm-sdk pod.

   b. Parse the **custom_flds.h** file and generate a **custom_flds.bat** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

c. Layer the *BRM_home*/**lib/custom_flds.bat** file in the following images: brm-apps, pdc, and cm.

   For example, to layer the file in the pdc image:

   ```
   FROM pdc:16.26.0.x.x.x

   USER root
   COPY custom_flds.bat ${PIN_HOME}/lib/custom_flds.bat
   RUN chown=oracle:root ${PIN_HOME}/lib/custom_flds.bat
   ```

d. In your brm-apps-2 ConfigMap (**configmap_pin_conf_brm_apps_2.yaml**), add the following entry under the **load_config.conf** section:

   ```
   load_config.conf: |
   # Making custom fields entry
      - - ops_fields_extension_file ${PIN_HOME}/lib/custom_flds.h
   ```

e. In your CM ConfigMap (**configmap_pin_conf_cm.yaml**), add the following entry under the **pin.conf** section:

   ```
   data:
      pin.conf:
      # Making custom fields entry
      - - ops_fields_extension_file ${PIN_HOME}/lib/custom_flds.h
   ```

f. In your **testnap** ConfigMap (**configmap_pin_conf_testnap**), add the following entry under the **pin.conf** section:

   ```
   data:
      pin.conf:  |
      # Making custom fields entry
      - - ops_fields_extension_file ${PIN_HOME}/lib/custom_flds.h
   ```

g. In your **override-values.yaml** file for **oc-cn-helm-chart**, update the **imageTag** keys to point to the new cm, brm-apps and pdc images.

   > **Note**
   >
   > Skip this step if your **override-values.yaml** file does not already contain **imageTag** keys for the cm, brm-apps, and pdc images.

h. Run the **helm upgrade** command to update your Helm release:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --
   namespace BrmNameSpace
   ```

Building Pipeline Configuration Center Images

The Pipeline Configuration Center image extends the Fusion Middleware Infrastructure image by packaging its own installer **PipelineConfigurationCenter_16.26.0.*x*.*x*.*x*_generic.jar** file along with scripts and configurations.

To build your own image of Pipeline Configuration Center, you must have these base images ready. The **oc-cn-docker-files-16.26.0.*x*.*x*.*x*.tgz** package includes references to all Dockerfiles and scripts that are needed to build images of Pipeline Configuration Center. You can refer to them when building a Pipeline Configuration Center image in your own environment.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Pulling the Fusion Middleware Infrastructure Image

The Fusion Middleware Infrastructure Image is available on the Oracle Container Registry (https://container-registry.oracle.com/). This image is regularly updated with the latest security fixes. You can pull this image to your local system, where you will build other images, with the name **container-registry.oracle.com/middleware/fmw-infrastructure_cpu:12.2.1.4-jdk8-ol7**.

Building the Pipeline Configuration Center Image

To build the Pipeline Configuration Center image, do this:

1. Go to the **oc-cn-docker-files/ocpcc/pcc** directory.
2. Download the Oracle Communications Pipeline Configuration Center installation JAR file.
3. Copy **PipelineConfigurationCenter_16.26.0.*x*.*x*.*x*_generic.jar** to the current working directory (**oc-cn-docker-files/ocpcc/pcc**).
4. Build the Pipeline Configuration Center image by entering this command:

   ```
   podman build --format docker --tag oracle/pcc:16.26.0.x.x.x .
   ```

Building Billing Care Images

The Billing Care image extends the Linux image by packaging the application archive along with scripts and configurations.

To build your own image of Billing Care, you need the Linux and JRE images, available on the Oracle Container Registry (https://container-registry.oracle.com). These images are regularly updated with the latest security fixes. You can pull these images to your local system, where you will build other images, with the names:

- container-registry.oracle.com/os/oraclelinux:8
- container-registry.oracle.com/java/serverjre:8-oraclelinux8

The **oc-cn-docker-files-16.26.0.*x*.*x*.*x*.tgz** package includes references to all Dockerfiles and scripts that are needed to build images of Billing Care. You can refer to them when building a Billing Care image in your own environment.

Pulling the Fusion Middleware Infrastructure Image

The Fusion Middleware Infrastructure Image is available on the Oracle Container Registry (https://container-registry.oracle.com). This image is regularly updated with the latest security fixes. You can pull this image to your local system, where you will build other images, with the name **container-registry.oracle.com/middleware/fmw-infrastructure_cpu:14.1.2.0-jdk21-ol9**.

Building the Billing Care Image

To build the Billing Care image, do this:

1. Go to the **oc-cn-docker-files/ocbc/billing_care** directory.
2. Download the Oracle Communications Billing Care installation JAR file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3. Copy **BillingCare_generic.jar** to the current working directory (**oc-cn-docker-files/ocbc/billing_care**).
4. Build the Billing Care image by entering this command:

   ```
   podman build --format docker --tag oracle/billingcare:16.26.0.x.x.x .
   ```

Building Business Operations Center Images

The Business Operations Center image extends the Linux image by packaging the application archive along with scripts and configurations.

To build your own image of Business Operations Center, you need the Linux and JRE images, available on the Oracle Container Registry (https://container-registry.oracle.com). These images are regularly updated with the latest security fixes. You can pull these images to your local system, where you will build other images, with the names:

- container-registry.oracle.com/os/oraclelinux:8
- container-registry.oracle.com/java/serverjre:8-oraclelinux8

The **oc-cn-docker-files-16.26.0.*x*.*x*.*x*.tgz** package includes references to all of the Dockerfiles and scripts needed to build images of Business Operations Center. You can refer to them when building a Business Operations Center image in your own environment.

To build the Business Operations Center image, do this:

1. Go to the **oc-cn-docker-files/ocboc/boc** directory.
2. Download the Oracle Communications Business Operations Center installation JAR file.
3. Copy **BusinessOperationsCenter_generic.jar** to the current working directory (**oc-cn-docker-files/ocboc/boc**).
4. Build the Business Operations Center image by entering this command:

   ```
   podman build --format docker --tag oracle/boc:16.26.0.x.x.x .
   ```

Building Collections Configuration Center Images

The Collections Configuration Center image extends the Linux image by packaging the application archive along with scripts and configurations.

To build your own image of Collections Configuration Center, you need the Linux and JRE images, available on the Oracle Container Registry (https://container-registry.oracle.com). These images are regularly updated with the latest security fixes. You can pull these images to your local system, where you will build other images, with the names:

- container-registry.oracle.com/os/oraclelinux:8
- container-registry.oracle.com/java/serverjre:8-oraclelinux8

The **oc-cn-docker-files-16.26.0.*x*.*x*.*x*.tgz** package includes references to all of the Dockerfiles and scripts needed to build images of Collections Configuration Center. You can refer to them when building a Collections Configuration Center image in your own environment.

To build the Collections Configuration Center image, do this:

1. Go to the **occcc/brm-collections-configuration-center/ccc** directory.
2. Download the Oracle Communications Collections Configuration Center installation JAR file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3. Copy **collections-configuration-center.jar** to the current working directory (**oc-cn-docker-files/occcc/brm-collections-configuration-center/ccc**).
4. Build the Collections Configuration Center image by entering this command:

   ```
   podman build --format docker --tag oracle/ccc:16.26.0.x.x.x .
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Part V

Upgrading BRM Cloud Native

This part provides information about upgrading your Oracle Communications Billing and Revenue Management (BRM) cloud native environment to the latest patch set or interim patch release. It contains the following chapters:

- Upgrading Your BRM Cloud Native Environment
- Performing Zero-Downtime Upgrades
- Performing Zero-Downtime Upgrades of Disaster Recovery Cloud Native Systems
- Migrating from On-Premise BRM to BRM Cloud Native

Copyright © 2019, 2026, Oracle and/or its affiliates.

17

Upgrading Your BRM Cloud Native Environment

Learn how to upgrade your Oracle Communications Billing and Revenue Management (BRM) cloud native environment to the latest release.

Topics in this document:

- Tasks for the BRM Cloud Native Upgrade

In this document, the BRM release currently running on your production system is called the *old* release. The release you are upgrading to is called the *new* release. For example, if you upgrade from BRM 12.0.0.*x*.0 to BRM 16.26.0, 12.0.0.*x*.0 is the old release and 16.26.0 is the new one.

Tasks for the BRM Cloud Native Upgrade

This section provides a list of tasks required to upgrade your BRM cloud native deployment to the latest release, patch set, or interim patch release. All patch sets and interim patches are cumulative, so they include the fixes from previous patch sets and interim patches. You can perform a direct upgrade from one patch set to another. For example, you can perform a direct upgrade of BRM cloud native from 12.0.0.*x*.0 to 16.26.0.

To upgrade your BRM cloud native deployment, complete these tasks in the specified order:

1. If you are upgrading from 12.0.0.2.0 to 12.0.0.3.0 or later, migrate your BRM cloud native Helm charts to the v3.*x* format by using the **helm2to3** utility. The Helm charts in BRM cloud native 12.0.0.2.0 use Helm v2.*x*, and later patch set releases use Helm v3.*x*. Helm v3.*x* doesn't readily understand the releases created by Helm v2.*x*.

   For more information, see "Migrating Helm v2 to v3" in the Helm documentation. The documentation contains references to the migration plugin and to a blog with a comprehensive walk-through of steps using a sample chart.

2. Upgrade your BRM cloud native database schema. See "Upgrading Your Database Schema".
3. Upgrade your BRM cloud native services. See "Upgrading Your BRM Cloud Native Services".
4. Upgrade your Elastic Charging Engine (ECE) cloud native services. See "Upgrading Your ECE Cloud Native Services".
5. Upgrade your client application services in any order:
   - Upgrade your Pricing Design Center (PDC) service and database schema. See "Upgrading Your PDC Cloud Native Services".
   - Upgrade your BRM REST Services Manager service. See "Upgrading BRM REST Services Manager".
   - Upgrade your Business Operations Center service and database schema. See "Upgrading Your Business Operations Center Cloud Native Services".

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Upgrade your Pipeline Configuration Center (PCC) service. See "Upgrading Your PCC Cloud Native Services".
- Upgrade your Billing Care and Billing Care REST API services. See "Upgrading Your Billing Care and Billing Care REST API Cloud Native Services".

Upgrading Your Database Schema

To upgrade your BRM cloud native database schema to the 16.26.0 release:

> **Note**
>
> BRM does not support a direct database schema upgrade from BRM 12.0.0.*x*.0 to BRM 16.26.0. To upgrade from BRM 12.0.0.*x*.0, first upgrade the database schema to BRM 15.2. After you complete the BRM 15.2 upgrade, follow this procedure to upgrade the database schema to BRM 16.26.0.

1. Download Oracle Communications Cloud Native Database Initializer Helm Chart 16.26.0 from the Oracle Software Delivery Cloud website. See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Download and load the BRM 16.26.0 cloud native component images in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Extract the BRM database initializer Helm chart from the archive. For example:

   ```
   tar xvzf oc-cn-init-db-helm-chart-16.26.0.x.x.x.tgz
   ```

   If you are extracting an interim patch, the file name will also have the interim patch number appended to it, such as **oc-cn-init-db-helm-chart-16.26.0.0.0.0-12345678.tgz**.
4. Create an **override-init-db-16260.yaml** file for the 16.26.0 version of **oc-cn-init-db-helm-chart**.
5. In the **override-init-db-16260.yaml** file, do the following:
   - Set the **ocbrm.is_upgrade** key to **true**.
   - Configure the database connection information and credentials for the existing BRM database.
   - Set the other keys in Table 6-1 as required for your environment.

   BRM stores the root-key wallet in the existing BRM database and retrieves it during the database schema upgrade. You do not provide or migrate a separate root-key wallet to the database initializer Helm chart.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you configure an external Kubernetes secret for the database TLS wallet, create the secret in the upgrade namespace before you run the database initializer Helm chart. You do not create a separate Kubernetes secret for the BRM root-key wallet.

6. Validate the 16.26.0 **oc-cn-init-db-helm-chart** chart's content by using the **helm lint** command.
   - For Helm 3.6.0 and later releases, enter this command from the **helmcharts** directory:

     ```
     helm lint --strict oc-cn-init-db-helm-chart --values oc-cn-init-db-helm-chart/
     values.yaml --values override-init-db-16260.yaml
     ```

   - For earlier Helm releases, enter:

     ```
     helm lint --strict oc-cn-init-db-helm-chart
     ```

   You'll see this if the command completes successfully:

   ```
   1 chart(s) linted, 0 chart(s) failed
   ```

7. Upgrade the database schema by entering the following command from the **helmcharts** directory. Run the Helm chart with a new release name and namespace.

   ```
   helm install newRelease oc-cn-init-db-helm-chart --namespace newNameSpace --values
   override-init-db-16260.yaml
   ```

   where:
   - *newRelease* is the release name for your new BRM 16.26.0 database upgrade. It must be different from the release name of your existing deployment.
   - *newNameSpace* is the namespace in which to create the Kubernetes objects for the BRM 16.26.0 database upgrade. It must be different from the namespace of your existing deployment.

8. Verify that the database schema upgrade completed successfully:

   ```
   kubectl --namespace newNameSpace get pods
   ```

   If the upgrade succeeds, the upgrade pod has a status of **Completed**. For example:

   ```
   NAME           READY   STATUS      RESTARTS   AGE
   upgrade-wc6sx  0/1     Completed   0          22h
   ```

You have now upgraded your BRM cloud native database schema to BRM 16.26.0.

Upgrading Your BRM Cloud Native Services

> **Note**
>
> The steps for upgrading your BRM cloud native services are the same for old and new schemas.

When you upgrade your BRM cloud native services, it upgrades all BRM core services in your BRM cloud native environment.

Copyright © 2019, 2026, Oracle and/or its affiliates.

To upgrade your BRM cloud native services to the 16.26.0 release:

1. Download and install Oracle Communications Cloud Native Helm Chart 16.26.0 from the Oracle Software Delivery Cloud website. See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Download and load the BRM 16.26.0 cloud native images in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Extract the BRM 16.26.0 Helm chart from the archive:

   ```
   tar xvzf oc-cn-helm-chart-16.26.0.x.x.x.tgz
   ```

   If you are extracting an interim patch, the file name will also have the interim patch number appended to it, such as **oc-cn-helm-chart-16.26.0.*x.x.x*-12345678.tgz**.
4. Create an **upgrade-brm-15.yaml** file for the 16.26.0 version of **oc-cn-helm-chart**.
5. In the **upgrade-brm-15.yaml** file, set the following keys:
   - Set the **ocbrm.is_upgrade** key to **true**.
   - Set the other keys in Table 7-5 as needed.
6. Run the **helm upgrade** command for the 16.26.0 version of **oc-cn-helm-chart** using the same release name and namespace that you used for your old release:

   ```
   helm upgrade oldBrmReleaseName oc-cn-helm-chart --values oldOverrideValues --values
   upgrade-brm-15.yaml --namespace oldBrmNamespace
   ```

   where:
   - *oldBrmReleaseName* is the release name assigned to your old **oc-cn-helm-chart** installation.
   - *oldOverrideValues* is the file name and path to the **override-values.yaml** file for your old BRM installation.
   - *oldBrmNamespace* is the same namespace as for your old BRM deployment.
7. If you are upgrading a multischema system, do the following:
   a. Add the following lines to the 16.26.0 version of the **oc-cn-helm-chart/brmapps_scripts/loadme.sh** script:

      ```
      #!/bin/sh
      cd /oms/setup/scripts; perl pin_amt_install.pl -m
      exit 0;
      ```

   b. Enable the brm-apps job by setting these keys in your **override-values.yaml** file for the 16.26.0 version of **oc-cn-helm-chart**:
      - **ocbrm.brm_apps.job.isEnabled**: Set this to **true**
      - **ocbrm.brm_apps.job.isMultiSchema**: Set this to **false**
   c. Run the **helm upgrade** command for **oc-cn-helm-chart**:

      ```
      helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --
      namespace BrmNameSpace
      ```

   d. Restart the amt pod.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Your BRM core services have been upgraded to the latest 16.26.0 release.

8. After the upgrade completes successfully, restart your WebLogic Server Administration Server and Managed Servers by following the instructions in https://oracle.github.io/weblogic-kubernetes-operator/faq/domain-secret-mismatch/.

> **Note**
>
> The first time you run **pin_virtual_time** after upgrading the BRM core services to the 16.26.0 release, it generates a new 64-bit **pin_virtual_time_file** utility. You must restart the CM after the 64-bit **pin_virtual_time_file** is created. To do so:
>
> 1. In the **override-values.yaml** for **oc-cn-helm-chart**, set the following keys:
>   - **ocbrm.config_jobs.restart_count**: Increment the value by 1.
>   - **ocbrm.config_jobs.run_apps**: Set this to **true**.
> 2. Run the **helm upgrade** command for **oc-cn-helm-chart** using the same release name and namespace that you used for your old release:
>
>    ```
>    helm upgrade oldBrmReleaseName oc-cn-helm-chart --values oldOverrideValues --values upgrade-brm.yaml --namespace oldBrmNamespace
>    ```

Upgrading Your ECE Cloud Native Services

> **Note**
>
> The current ECE cloud native deployment uses a single ecs StatefulSet for ECE server pods. The ecs1 StatefulSet was used in earlier deployments but has been removed from current deployments.

To upgrade your ECE cloud native services to release 16.26.0:

1. Download the ECE cloud native Helm chart. See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Extract the Helm chart into a separate staging area. For example:

   ```
   tar xvzf oc-cn-ece-helm-chart-16.26.0.x.x.x.tgz StagingArea
   ```

3. In the staging area, create a diff between the new **values.yaml** file and your existing **oc-cn-ece-helm-chart/values.yaml** file.

   For example, if upgrading from 12.0.0.5.0 to 16.26.0, compare the **values.yaml** files from both versions.
4. Using the diff, identify keys that were added, changed, or removed in the new release.
5. Open your existing **override-values.yaml** file for **oc-cn-ece-helm-chart**. This file contains your previous customizations.
6. Apply the necessary changes:
   - Add and configure any new keys you want to use.
   - Remove keys that are no longer supported.
   - Review keys with changed default values and decide whether to update them.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> For live-network upgrades, set **charging.rebuildCdrStoreTable** to **false** in your **override-values.yaml** file for **oc-cn-ece-helm-chart**. This setting prevents the ECE persistence upgrade job from dropping and re-creating the **cdrstore** table and its indexes during the upgrade. Keeping this key set to **false** helps avoid upgrade failures when ECE traffic creates active transactions or database locks on the **cdrstore** table. If the persistence upgrade job attempts to drop the table while it is locked, the **ece-persistence-upgrade-job** can fail. This setting also helps preserve existing CDR data during the upgrade.

7. Ensure that at least three ecs pod replicas are configured:
   - **charging.ecs.replicas**: Set this to a value of 3 or greater.
8. If your current ECE cloud native deployment does not support cdrstore tablespaces and you are upgrading to a version that requires them, do one of the following:
   - Continue without separate tablespaces for cdrstore. To do so, in your **override-values.yaml** file, set the **cdrstoretablespace** and **cdrstoreindexspace** keys to an empty value:

     ```
     cdrstoretablespace: ""
     cdrstoreindexspace: ""
     ```

   - Use separate tablespaces for cdrstore. To do so, in **override-values.yaml** file, set the **cdrstoretablespace** and **cdrstoreindexspace** keys to the tablespace names.

     Also, grant quota on the new tablespaces to the ECE schema user. Connect to your database as the **system** user using SQL*Plus, and enter these commands:

     ```
     SQL> ALTER USER EceSchemaUser quota unlimited on ECECDRTABLESPACE;
     SQL> ALTER USER EceSchemaUser quota unlimited on ECECDRINDEXSPACE;
     ```

9. Save and close your **override-values.yaml** file to your staging area.
10. Delete all old ECE template files from your staging area's **oc-cn-ece-helm-chart/templates** directory.
11. Copy the new ECE template files to your staging area's **oc-cn-ece-helm-chart/templates** directory.
12. Upgrade your ECE cloud native services by running these commands:

    ```
    cd StagingArea/oc-cn-ece-helm-chart/
    sh upgradeECE_16.26.0.x.x.x.sh -o OverrideValuesFile -n BrmNameSpace -r
    EceReleaseName -s y
    ```

    where:

    - *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **oc-cn-ece-helm-chart/values.yaml** file.
    - *BrmNameSpace* is the namespace in which BRM Kubernetes objects reside for the BRM Helm chart.
    - *EceReleaseName* is the release name for **oc-cn-ece-helm-chart** and is used to track this installation instance. It must be different from the one used for the BRM Helm chart.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> The **upgradeECE_16.26.0.*x*.*x*.*x*.sh** script performs sanity checks, controlled scale-down of ecs, label application, and other steps required to prevent cache-loss scenarios.

Upgrading ECE Cloud Native to the Latest Interim Patch

To upgrade your ECE cloud native services from 16.26.0 to the latest 16.26.0 interim patch:

1. Delete any existing ECE Kubernetes jobs:

   ```
   kubectl --namespace BrmNameSpace get job
   kubectl --namespace BrmNameSpace delete job JobName
   ```

   where *BrmNameSpace* is the namespace in which BRM Kubernetes objects reside for the BRM Helm chart, and *JobName* is the name of the Kubernetes job.
2. Download the latest 16.26.0 interim patch release from the My Oracle Support website (https://support.oracle.com).
3. Extract the interim patch's ECE Helm chart from the archive into a separate staging area. For example:

   ```
   tar xvzf oc-cn-ece-helm-chart-16.26.0.0.0.xyz.tgz StagingArea
   ```

   where *xyz* is the interim patch number.
4. In your staging area, create a diff between the 16.26.0 interim patch **values.yaml** file and your old 16.26.0 **oc-cn-ece-helm-chart/values.yaml** file.
5. Using the diff, make a list of the keys that were added, changed, and removed in the 16.26.0 interim patch release.
6. Open your 16.26.0 **override-values.yaml** file for **oc-cn-ece-helm-chart**.
7. Do the following:
   - Add and configure any new keys that you want to use.
   - Delete the keys that were removed.
   - If a key's default value changed, determine whether you want to override the key's value.

   > **Note**
   >
   > For live-network upgrades, set **charging.rebuildCdrStoreTable** to **false** in your **override-values.yaml** file for **oc-cn-ece-helm-chart**. This setting prevents the ECE persistence upgrade job from dropping and re-creating the **cdrstore** table and its indexes during the upgrade. Keeping this key set to **false** helps avoid upgrade failures when ECE traffic creates active transactions or database locks on the **cdrstore** table. This setting also helps preserve existing CDR data during the upgrade.
8. Ensure that you have at least three ecs pod replicas configured in the file:
   - **charging.ecs.replicas**: Set this to a value of 3 or greater.
9. Save and close your **override-values.yaml** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

10. Upgrade your ECE cloud native services to the latest 16.26.0 interim patch release by running these commands:

    ```
    cd StagingArea/oc-cn-ece-helm-chart/
    sh upgradeECE_16.26.0.0.0.0.sh --override OverrideValuesFile --namespace
    BrmNameSpace --release EceReleaseName
    ```

    where:

    - *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **oc-cn-ece-helm-chart/values.yaml** file.
    - *EceReleaseName* is the release name for **oc-cn-ece-helm-chart** and is used to track this installation instance. It must be different from the one used for the BRM Helm chart.

Upgrading Your PDC Cloud Native Services

> **Note**
>
> - When you upgrade your PDC cloud native services, your PDC database is also upgraded.
> - If you reuse an old PDC schema, you must choose the same rating engine as the old PDC installation. That is, if your old PDC installation uses Elastic Charging Engine (ECE) for usage rating, you cannot switch to the Real-time Rating and Batch Rating Engines. Likewise, you cannot switch to ECE if your old PDC installation uses the Real-time Rating and Batch Rating Engines for usage rating. If you attempt to switch rating engines, the PDC upgrade fails and generates an error message.
> - You cannot migrate pricing data to PDC cloud native systems, because the PDC **MigrateBRMPricing** utility is not supported in cloud native environments.

To upgrade your PDC cloud native services and the PDC database to the latest 16.26.0 release:

1. Download and extract the BRM 16.26.0 cloud native Helm charts. See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Download and push the PDC 16.26.0 component images into your repository in one of these ways:
    - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
    - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Remove all PDC pods, services, and volume mounts from your PDC 12.0.0.*x*.0 release by doing the following:
    a. In your **override-values.yaml** file for the 12.0.0.*x*.0 version of **oc-cn-helm-chart**, set the **ocpdc.isEnabled** key to **false**.
    b. Run the **helm upgrade** command for the 12.0.0.*x*.0 release of **oc-cn-helm-chart**:

    ```
    helm upgrade oldBrmReleaseName oc-cn-helm-chart --values
    oldBrmOverrideValuesFile --namespace oldBrmNameSpace
    ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

where:

- *oldBrmReleaseName* is the release name assigned to your 12.0.0.*x*.0 version of the **oc-cn-helm-chart** installation.
- *oldBrmOverrideValuesFile* is the file name and path of your 12.0.0.*x*.0 version of the **override-values.yaml** file for **oc-cn-helm-chart**.
- *oldBrmNameSpace* is the namespace for your 12.0.0.*x*.0 version of the BRM deployment.

> **Note**
>
> Ensure that all PDC pods, services, and volume mounts have been deleted.

4. If you are upgrading from PDC 12.0.0.8.0 or later to PDC 16.26.0, do the following:
   a. In your **override-values.yaml** file for your *old release* of **oc-cn-op-job-helm-chart**, set the **ocpdc.isEnabled** key to **false**.
   b. Run the **helm upgrade** command for your *old release* of **oc-cn-op-job-helm-chart**:

      ```
      helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
      oldOpJobOverrideValuesFile --namespace oldBrmNameSpace
      ```

      where:

      - *oldOpJobReleaseName* is the release name assigned to your old release of the **oc-cn-op-job-helm-chart** installation.
      - *oldOpJobOverrideValuesFile* is the file name and path of your old release of the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.
      - *oldBrmNameSpace* is the namespace for your old BRM deployment.

      Ensure all PDC jobs are deleted.
   c. Compare your old release's **oc-cn-op-job-helm-chart/values.yaml** file with the 16.26.0 version of that file.
      - Create a diff between your old release's **values.yaml** file and the 16.26.0 release's **values.yaml** file.
      - Using the diff, make a list of the keys that were added, changed, and removed in the 16.26.0 release.
      - Open *oldOpJobOverrideValuesFile*. This file contains all of the customizations that you made in previous releases.
      - Do the following:
        – Add and configure any new keys that you want to use.
        – Delete the keys that were removed.
        – If a key's default value changed, determine whether you want to modify the key's value.
      - Close and save the file as **new-op-job-override-values.yaml**.
5. Do one of the following:
   - If you are upgrading from 12.0 Patch Set 8 or later, open your **new-op-job-override-values.yaml** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- If you are upgrading from 12.0 Patch Set 7 or earlier, create an **override-values.yaml** file for the 16.26.0 version of **oc-cn-op-job-helm-chart** named **new-op-job-override-values.yaml**.

6. In **new-op-job-override-values.yaml**, set the following keys:
   - **ocpdc.isEnabled**: Set this to **true**.
   - **ocpdc.configEnv.upgrade**: Set this to **true**.
   - **ocpdc.configEnv.siteNo**: Set this to **1**.
7. Run the **helm upgrade** command for the 16.26.0 version of **oc-cn-op-job-helm-chart**:

   ```
   helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values new-op-job-override-values.yaml --namespace oldBrmNameSpace
   ```

   Wait for the PDC domain job to complete.
8. Compare your 12.0.0.*x*.0 release's **oc-cn-helm-chart/values.yaml** file with the 16.26.0 version of the file.
   - Create a diff between your old release's **values.yaml** file and the 16.26.0 release's **values.yaml** file.
   - Using the diff, make a list of the keys that were added, changed, and removed in the 16.26.0 release.
   - Open your 12.0.0.*x*.0 release's **override-values.yaml** file. This file contains all of the customizations that you made in previous releases.
   - Do the following:
     - Add and configure any new keys that you want to use.
     - Delete the keys that were removed.
     - If a key's default value changed, determine whether you want to modify the key's value.
   - Close and save the file as **new-brm-override-values.yaml**.
9. In **new-brm-override-values.yaml**, set the following keys:
   - **ocbrm.pdc_deployed**: Set this to **true**.
   - **ocpdc.configEnv.upgrade**: Set this to **true**.

     This value must match the one set in your **override-values.yaml** file for **oc-cn-op-job-helm-chart** (in Step 6).
   - **ocpdc.isEnabled**: Set this to **true**.
10. Run the **helm upgrade** command for the 16.26.0 version of **oc-cn-helm-chart**:

    ```
    helm upgrade oldBrmReleaseName oc-cn-helm-chart --values new-brm-override-values.yaml --namespace oldBrmNameSpace
    ```

Upgrading BRM REST Services Manager

To upgrade your BRM REST Services Manager cloud native services to the 16.26.0 release:

1. Download and extract the BRM cloud native Helm charts. See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".

Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Download and push the BRM REST Services Manager component image to your repository in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling WebLogic Images for PDC, Billing Care, Billing Care REST API, and Business Operations Center".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Disable the brm-rest-services-manager service in your BRM cloud native environment.
   a. Create an **upgrade-brm-rsm.yaml** file. This file will be used by **oc-cn-helm-chart**.
   b. In your **upgrade-brm-rsm.yaml** file, set the **ocrsm.rsm.isEnabled** key to **false**.
   c. Stop the brm-rest-services-manager pod by running the Helm upgrade command for the **oc-cn-helm-chart**:

      ```
      helm upgrade oldBrmReleaseName oc-cn-helm-chart --values oldOverrideValues --
      values upgrade-brm-rsm.yaml --namespace oldBrmNamespace
      ```

      where:
      - *oldBrmReleaseName* is the BRM release name for your old release.
      - *oldOverrideValues* is the file name and path to the **override-values.yaml** file for your old **brm-rest-services-manager** installation.
      - *oldBrmNamespace* is the BRM namespace for your old release.
4. Wait for the brm-rest-services-manager pod to stop.
5. In your **upgrade-brm-rsm.yaml** file, set the these keys:
   - **ocrsm.rsm.isEnabled**: Set this to **true**.
   - **ocrsm.rsm.deployment.imageTag**: Set this to the new release number in the format 16.26.0.0.0.0 or 16.26.0.0.0.0-*nnnnnnnn* for interim patches.
6. Copy the SSL Certificate for BRM REST Services Manager.
   a. Create a directory named **rsm_keystore** under the newly extracted **oc-cn-helm-chart/rsm** directory.
   b. Copy the files created in the step "Generating an SSL Certificate for BRM REST Services Manager" to the newly created **oc-cn-helm-chart/rsm** directory.
   c. Start your brm-rest-services-manager services by running the Helm upgrade command for **oc-cn-helm-chart**:

      ```
      helm upgrade oldBRMReleaseName oc-cn-helm-chart --values oldOverrideValues --
      values upgrade-brm-rsm.yaml --namespace oldBrmNamespace
      ```

Upgrading Your Business Operations Center Cloud Native Services

The instructions to upgrade your Business Operations Center services differ depending on the patch set you are upgrading to or from.

- To upgrade the Business Operations Center service from 12.0.0.7.0 or earlier to 16.26.0, follow the instructions in "Upgrading Your Business Operations Center Cloud Native Service from 12.0.0.7.0 or Earlier to 16.26.0".
- To upgrade the Business Operations Center service from 12.0.0.8.0 to 16.26.0, follow the instructions in "Upgrading Your Business Operations Center Cloud Native Service from 12.0.0.8.0 to 16.26.0".

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> When you upgrade your Business Operations Center cloud native service, you can also upgrade your Business Operations Center database schema.

Upgrading Your Business Operations Center Cloud Native Service from 12.0.0.7.0 or Earlier to 16.26.0

To upgrade your Business Operations Center cloud native service and database schema from 12.0.0.7.0 or earlier to 16.26.0:

1. Download and extract the 16.26.0 versions of the BRM cloud native Helm chart (**oc-cn-helm-chart**) and the cloud native operator job chart (**oc-cn-op-job-helm-chart**).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".

2. Download and push the Business Operations Center component image (**boc**) to your repository in one of these ways:

   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".

3. Disable all Business Operations Center 12.0.0.*x*.0 services in your BRM cloud native environment.

   a. Create an **upgrade-boc.yaml** file and then set the **ocboc.boc.isEnabled** key to **false**.

      The **upgrade-boc.yaml** file will be used with both **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.

   b. Stop the WebLogic domain by running the **helm upgrade** command for **oc-cn-helm-chart**:

      ```
      helm upgrade oldBrmReleaseName oc-cn-helm-chart --values oldOverrideValues --
      values upgrade-boc.yaml --namespace oldBrmNamespace
      ```

      where:

      - *oldBrmReleaseName* is the BRM release name for your old release.
      - *oldOverrideValues* is the **override-values.yaml** file for your 12.0.0.*x*.0 release.
      - *oldBrmNamespace* is the BRM namespace for your old release.

   c. Remove the WebLogic domain by running the **helm upgrade** command for **oc-cn-op-job-helm-chart**:

      ```
      helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
      oldOverrideValues --values upgrade-boc.yaml --namespace oldBrmNameSpace
      ```

      where *oldOpJobReleaseName* is the **oc-cn-op-job-helm-chart** release name for the old release.

4. Clean up the data in your Business Operations Center 12.0.0.*x*.0 persistent volumes (PVs).

   a. Clean up the domain home from the PV for Business Operations Center 12.0.0.*x*.0:

      ```
      rm -rf Domain_home/domains/domainUID
      ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

where:

- *Domain_home* is the location specified in the **ocboc.boc.wop.domainVolHostPath** key.
- *domainUID* is the domain name specified in the **ocboc.boc.wop.domainUID** key. The default is **boc-domain**.

See Table 9-1 for more information.

b. Clean up the application home from the PV for Business Operations Center:

```
rm -rf Application_home/BOC
```

where *Application_home* is the path specified in the **ocboc.boc.wop.appVolHostPath** key.

5. Compare your old 12.0.0.*x*.0 versions of the **oc-cn-helm-chart/values.yaml** and **oc-cn-op-job-helm-chart/values.yaml** files with the 16.26.0 versions of those files.
   - Create a diff between the 16.26.0 **values.yaml** file and your old 12.0.0.*x*.0 **values.yaml** file.
   - Using the diff, make a list of the keys that were added, changed, and removed in the new release.
   - Open your old 12.0.0.*x*.0 release's **override-values.yaml** file. This file contains all of the customizations that you made in previous releases.
   - Do the following:
     – Add and configure any new keys that you want to use.
     – Delete the keys that were removed.
     – If a key's default value changed, determine whether you want to modify the key's value.
   - Close and save the files as *updatedoldOverrideValues*.
6. Deploy Business Operations Center 16.26.0 with the latest changes and upgrade the Business Operations Center database schema.
   a. In your **upgrade-boc.yaml** file, set these Business Operations Center keys:
      - **ocboc.boc.isEnabled**: Set this to **true**.
      - **ocboc.boc.deployment.imageTag**: Set this to the new release number in the format 16.26.0.0.*x*.0 for patch sets and 16.26.0.0.*x*.0-*nnnnnnn* for interim patches.
   b. Run the **helm upgrade** command for **oc-cn-op-job-helm-chart**:

```
helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
updatedoldOverrideValues --values upgrade-boc.yaml --namespace oldBrmNameSpace
```

Wait for the jobs to complete their tasks.

7. Start your Business Operations Center 16.26.0 services by running the **helm upgrade** command for **oc-cn-helm-chart**:

```
helm upgrade oldBRMReleaseName oc-cn-helm-chart --values updatedoldOverrideValues --
values upgrade-boc.yaml --namespace oldBrmNamespace
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Upgrading Your Business Operations Center Cloud Native Service from 12.0.0.8.0 to 16.26.0

To upgrade your Business Operations Center cloud native service and database schema from 12.0.0.8.0 to 16.26.0:

1. Download and extract the 16.26.0 versions of the BRM cloud native Helm chart (**oc-cn-helm-chart**) and the cloud native operator job chart (**oc-cn-op-job-helm-chart**).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".

2. Download and push the Business Operations Center 16.26.0 image (**boc**) to your repository in one of these ways:

   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".

3. Compare your old 12.0.0.8.0 versions of the **oc-cn-helm-chart/values.yaml** and **oc-cn-op-job-helm-chart/values.yaml** files with the 16.26.0 versions of those files.

   - Create a diff between the 16.26.0 **values.yaml** file and your old 12.0.0.8.0 **values.yaml** file.
   - Using the diff, make a list of the keys that were added, changed, and removed in the new release.
   - Open your old 12.0.0.8.0 release's **override-values.yaml** file. This file contains all of the customizations that you made in previous releases.
   - Do the following:
     - Add and configure any new keys that you want to use.
     - Delete the keys that were removed.
     - If a key's default value changed, determine whether you want to modify the key's value.
   - Close and save the files as *updatedOldOverrideValues*.

4. Deploy Business Operations Center 16.26.0 with the latest changes and upgrade the Business Operations Center database schema.

   a. In your **upgrade-boc.yaml** file, set the **ocboc.boc.deployment.app.imageTag** key to the new release number in the format 16.26.0.0.0.0 or 16.26.0.0.0.0-*nnnnnnnn* for interim patches.

      The **upgrade-boc.yaml** file will be used with both **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.

   b. Run the **helm upgrade** command for **oc-cn-op-job-helm-chart**:

      ```
      helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
      updatedOldOverrideValues --values upgrade-boc.yaml --namespace oldBrmNameSpace
      ```

      Wait for the jobs to complete their tasks.

5. Start your Business Operations Center 16.26.0 services by running the **helm upgrade** command for **oc-cn-helm-chart**:

   ```
   helm upgrade oldBRMReleaseName oc-cn-helm-chart --values updatedOldOverrideValues --
   values upgrade-boc.yaml --namespace oldBrmNamespace
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Upgrading Your PCC Cloud Native Services

The instructions to upgrade your PCC cloud native services differ depending on the release you are upgrading from.

- To upgrade your PCC cloud native services from 12.0.0.*x*.0 or earlier to 15.0.0.0.0, follow the instructions in "Upgrading Your PCC Cloud Native Services from 12.0.0.*x*.0 or Earlier to 15.0.0.0.0".
- To upgrade your PCC cloud native services from 15.0 to 16.26.0, follow the instructions in "Upgrading Your PCC Cloud Native Services from 15.0 to 16.26.0".

Upgrading Your PCC Cloud Native Services from 12.0.0.*x*.0 or Earlier to 15.0.0.0.0

To upgrade your PCC cloud native services from 12.0.0.*x*.0 or earlier to the 15.0.0.0.0 release:

1. Download and extract the BRM 15.0.0.0.0 cloud native Helm chart (**oc-cn-helm-chart**).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".

2. Download and push the PCC 15.0.0.0.0 component image (**oracle/pcc:15.0.0.0.0**) to your repository in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Disable the PCC 12.0.0.*x*.0 cloud native services in your BRM cloud native environment:
   a. Create an **upgrade-pcc.yaml** file for **oc-cn-helm-chart**.
   b. In your **upgrade-pcc.yaml** file, set the **ocpcc.pcc.isEnabled** key to **false**.
   c. Stop the running pcc pod by running the **helm upgrade** command for the 12.0.0.*x*.0 version of **oc-cn-helm-chart**:

      ```
      helm upgrade oldBrmReleaseName oc-cn-helm-chart --values oldOverrideValues --
      values upgrade-pcc.yaml --namespace oldBrmNamespace
      ```

      where:
      - *oldBrmReleaseName* is the BRM release name for your 12.0.0.*x*.0 release.
      - *oldOverrideValues* is the file name and path to the **override-values.yaml** file for your 12.0.0.*x*.0 BRM installation.
      - *oldBrmNamespace* is the BRM namespace for your 12.0.0.*x*.0 release.
4. Wait for the pcc pod to stop.
5. Copy the SSL Certificate for PCC:
   a. Create a **keystore_pcc** directory under **oc-cn-helm-chart/pcc**.
   b. Copy the default PKCS12 certificate and KeyStore files to the **oc-cn-helm-chart/pcc** directory.

      During deployment, Helm uses the KeyStore files to create a Secret, which will be mounted as a volume inside the pcc pod.
   c. If your KeyStore files have file names different from what is specified in the **values.yaml** file, update the **keyStoreType**, **keyStoreIdentityFileName**, and **keyStoreTrustFileName** keys in your **override-values.yaml** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

6. Compare your 12.0.0.*x*.0 release's **oc-cn-helm-chart/values.yaml** file with the 15.0.0.0.0 version of the file.
   - Create a diff between your 12.0.0.*x*.0 release's **values.yaml** file and the 15.0.0.0.0 release's **values.yaml** file.
   - Using the diff, make a list of the keys that were added, changed, and removed in the 15.0.0.0.0 release.
   - Open your 12.0.0.*x*.0 release's **override-values.yaml** file. This file contains all of the customizations that you made in previous releases.
   - Do the following:
     - Add and configure any new keys that you want to use.
     - Delete the keys that were removed.
     - If a key's default value changed, determine whether you want to modify the key's value.
   - Close and save the file as **new-brm-override-values.yaml**.
7. In your **upgrade-pcc.yaml** file, set the following keys:
   - **ocpcc.pcc.isEnabled**: Set this to **true**.
   - **ocpcc.pcc.deployment.imageTag**: Set this to the new release number in the format 15.0.0.0.0 or 15.0.0.0.0-*nnnnnnnn* for interim patches.
8. Start your PCC cloud native services by running the **helm upgrade** command for the 15.0.0.0.0 version of **oc-cn-helm-chart**:

   ```
   helm upgrade oldBRMReleaseName oc-cn-helm-chart --values new-brm-override-values.yaml --values upgrade-pcc.yaml --namespace oldBrmNamespace
   ```

Upgrading Your PCC Cloud Native Services from 15.0 to 16.26.0

To upgrade your PCC cloud native services from 15.0 to 16.26.0:

1. Download and extract the BRM 15.0 cloud native Helm charts (**oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Download and push the PCC 16.26.0 component image (**pcc**) to your repository in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Disable all PCC 15.0 cloud native services in your BRM cloud native environment:
   a. Create an **upgrade-pcc.yaml** file and then set the **ocpcc.pcc.isEnabled** key to **false**.

      The **upgrade-pcc.yaml** file will be used with both **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.

   b. Stop the WebLogic domain by running the **helm upgrade** command for the 15.0 version of **oc-cn-helm-chart**:

      ```
      helm upgrade oldBrmReleaseName oc-cn-helm-chart --values oldOverrideValues --values upgrade-pcc.yaml --namespace oldBrmNamespace
      ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

where:

- *oldBrmReleaseName* is the BRM release name for your 15.0 release.
- *oldOverrideValues* is the **override-values.yaml** file for your 15.0 release.
- *oldBrmNamespace* is the BRM namespace for your 15.0 release.

c. Remove the WebLogic domain by running the **helm upgrade** command for the 15.0 version of **oc-cn-op-job-helm-chart**:

```
helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
oldOverrideValues --values upgrade-pcc.yaml --namespace oldBrmNameSpace
```

where *oldOpJobReleaseName* is the **oc-cn-op-job-helm-chart** release name for the 15.0 release.

4. Clean up the data in your PCC 15.0 persistent volumes (PVs).

a. Clean up the domain home from the PV for PCC 15.0:

```
rm -rf Domain_home/domains/domainUID
```

where:

- *Domain_home* is the location specified in the **ocpcc.pcc.wop.domainVolHostPath** key.
- *domainUID* is the domain name specified in the **ocpcc.pcc.wop.domainUID** key. The default is **pcc-domain**.

b. Clean up the application home from the PV for PCC 15.0:

```
rm -rf Application_home/PCC
```

where *Application_home* is the path specified in the o**cpcc.pcc.wop.appVolHostPath** key.

5. Compare your 15.0 version of the **oc-cn-helm-chart/values.yaml** file with the 16.26.0 version of that file.

- Create a diff between the 16.26.0 **values.yaml** file and the 15.0 **values.yaml** file.
- Using the diff, make a list of the keys that were added, changed, and removed in the 16.26.0 release.
- Open your 15.0 release's **override-values.yaml** file. This file contains all of the customizations that you made in previous releases.
- Do the following:
  - Add and configure any new keys that you want to use.
  - Delete the keys that were removed.
  - If a key's default value changed, determine whether you want to modify the key's value.
- Close and save the file as **new-brm-override-values.yaml**.

6. Configure the 16.26.0 version of **oc-cn-op-job-helm-chart**:

a. Create an **op-job-override-values.yaml** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

b. In the file, set the PCC-related keys for **oc-cn-op-job-helm-chart**. See "Adding Pipeline Configuration Center Keys for oc-cn-op-job-helm-chart".

7. In your **upgrade-pcc.yaml** file, set these PCC keys:
   - **ocpcc.pcc.isEnabled**: Set this to **true**.
   - **ocpcc.pcc.deployment.imageTag**: Set this to the new release number in the format 16.26.0.*x*.0.0 for patch sets and 16.26.0.*x*.0.0-*nnnnnnnn* for interim patches.
8. Deploy PCC 16.26.0 with the latest changes by running the **helm upgrade** command for **oc-cn-op-job-helm-chart**:

   ```
   helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values op-job-override-values.yaml --values upgrade-pcc.yaml --namespace oldBrmNameSpace
   ```
9. Start your PCC 16.26.0 cloud native services by running the **helm upgrade** command for **oc-cn-helm-chart**:

   ```
   helm upgrade oldBRMReleaseName oc-cn-helm-chart --values new-brm-override-values.yaml --values upgrade-pcc.yaml --namespace oldBrmNamespace
   ```

Upgrading Your Billing Care and Billing Care REST API Cloud Native Services

The instructions to use to upgrade your Billing Care and Billing Care REST API services are different, depending on the patch set you are upgrading from.

- To upgrade Billing Care and Billing Care REST API services from 12.0 Patch Set 7 or earlier, follow the instructions in "Upgrading Your Billing Care and Billing Care REST API Cloud Native Services from 12.0.0.7.0 or Earlier to 16.26.0".
- To upgrade the Billing Care and Billing Care REST API services from 12.0 Patch Set 8 or later, follow the instructions in "Upgrading Your Billing Care and Billing Care REST API Cloud Native Services from 12.0.0.8.0 to 16.26.0".

Upgrading Your Billing Care and Billing Care REST API Cloud Native Services from 12.0.0.7.0 or Earlier to 16.26.0

To upgrade Billing Care and Billing Care REST API from 12.0.0.7.0 or earlier to 16.26.0:

1. Download and extract the 16.26.0 versions of the BRM cloud native Helm chart (**oc-cn-helm-chart**) and the cloud native operator job chart (**oc-cn-op-job-helm-chart**).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Download and push the 16.26.0 versions of the Billing Care image (**billingcare**) and the Billing Care REST API image (**bcws**) to your repository in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Disable all Billing Care and Billing Care REST API services in your BRM 12.0.0.*x*.0 cloud native environment.

   a. Create an **upgrade-billing.yaml** file.

      This file will be used with both **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.

   b. In your **upgrade-billing.yaml** file, set these keys:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **ocbc.bc.isEnabled**: Set this to **false**.
- **ocbc.bcws.isEnabled**: Set this to **false**.

c. Stop the WebLogic domain by running the Helm upgrade command for **oc-cn-helm-chart**:

```
helm upgrade oldBrmReleaseName oc-cn-helm-chart --values oldOverrideValues --
values upgrade-billing.yaml --namespace oldBrmNamespace
```

where:

- *oldBrmReleaseName* is the BRM release name for your old release.
- *oldOverrideValues* is the file name and path to the **override-values.yaml** file for your old Billing Care installation.
- *oldBrmNamespace* is the BRM namespace for your old release.

d. Remove the WebLogic domain by running the Helm upgrade command for **oc-cn-op-job-helm-chart**:

```
helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
oldOverrideValues --values upgrade-billing.yaml --namespace oldBrmNameSpace
```

where *oldOpJobReleaseName* is the **oc-cn-op-job-helm-chart** release name for the old release.

4. Clean up the data in the 12.0.0.*x*.0 versions of the Billing Care and Billing Care REST API persistent volumes (PVs).

a. Clean up the domain home from the PV for Billing Care and Billing Care REST API:

```
rm -rf Domain_home/domains/domainUID
```

where:

- *Domain_home* is the location specified in the **ocbc.bc.wop.domainVolHostPath** and **ocbc.bcws.wop.domainVolHostPath** keys.
- *domainUID* is the domain name specified in the **ocbc.bc.wop.domainUID** and **ocbc.bcws.wop.domainUID** keys. The defaults are **billingcare-domain** and **bcws-domain**.

See Table 9-4 and Table 9-6.

b. Clean up the application home from the PV for Billing Care and Billing Care REST API:

```
rm -rf Application_home/billingcare
```

where *Application_home* is the path specified in the **ocbc.bc.wop.appVolHostPath** and **ocbc.bcws.wop.appVolHostPath** keys.

5. Compare your old 12.0.0.*x*.0 versions of the **oc-cn-helm-chart/values.yaml** and **oc-cn-op-job-helm-chart/values.yaml** files with the 16.26.0 versions of those files.

- Create a diff between the 16.26.0 **values.yaml** file and your old 12.0.0.*x*.0 **values.yaml** file.
- Using the diff, make a list of the keys that were added, changed, and removed in the new release.
- Open your old 12.0.0.*x*.0 release's **override-values.yaml** file. This file contains all of the customizations that you made in previous releases.
- Do the following:
  - Add and configure any new keys that you want to use.

Copyright © 2019, 2026, Oracle and/or its affiliates.

  - Delete the keys that were removed.
  - If a key's default value changed, determine whether you want to modify the key's value.
- Close and save the files as *updatedOldOverrideValues*.

6. In your **upgrade-billing.yaml** file, set these Billing Care and Billing Care REST API keys:
   - **ocbc.bc.isEnabled**: Set this to **true**.
   - **ocbc.bcws.isEnabled**: Set this to **true**.
   - **ocbc.bc.deployment.imageTag**: Set this to the new release number in the format 16.26.0.*x.x.x* or 16.26.0.*x.x.x-nnnnnnnn* for interim patches.
   - **ocbc.bcws.deployment.imageTag**: Set this to the new release number in the format 16.26.0.*x.x.x* or 16.26.0.*x.x.x-nnnnnnnn* for interim patches.
7. Deploy the 16.26.0 versions of the Billing Care and Billing Care REST API with the latest changes by running the Helm upgrade command for **oc-cn-op-job-helm-chart**:

   ```
   helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
   updatedOldOverrideValues --values upgrade-billing.yaml --namespace oldBrmNameSpace
   ```

   Wait for the jobs to complete their tasks.
8. Start your 16.26.0 versions of the Billing Care and Billing Care REST API services by running the Helm upgrade command for **oc-cn-helm-chart**:

   ```
   helm upgrade oldBRMReleaseName oc-cn-helm-chart --values updatedOldOverrideValues --
   values upgrade-billing.yaml --namespace oldBrmNamespace
   ```

Upgrading Your Billing Care and Billing Care REST API Cloud Native Services from 12.0.0.8.0 to 16.26.0

To upgrade Billing Care and Billing Care REST API cloud native from the 12.0.0.8.0 release to 16.26.0:

1. Download and extract the 16.26.0 versions of the BRM cloud native Helm chart (**oc-cn-helm-chart**) and the cloud native operator job chart (**oc-cn-op-job-helm-chart**).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files".
2. Download and push the 16.26.0 versions of the Billing Care image (**billingcare**) and the Billing Care REST API image (**bcws**) to your repository in one of these ways:
   - From the Oracle Container Registry. To do so, see "Pulling BRM Images from the Oracle Container Registry".
   - From the Oracle Software Delivery website. To do so, see "Downloading BRM Images from Oracle Software Delivery Website".
3. Compare your old 12.0.0.8.0 versions of the **oc-cn-helm-chart/values.yaml** and **oc-cn-op-job-helm-chart/values.yaml** files with the 16.26.0 versions of those files.
   - Create a diff between the 16.26.0 **values.yaml** file and your old 12.0.0.8.0 **values.yaml** file.
   - Using the diff, make a list of the keys that were added, changed, and removed in the new release.
   - Open your old 12.0.0.8.0 release's **override-values.yaml** file. This file contains all of the customizations that you made in previous releases.
   - Do the following:

Copyright © 2019, 2026, Oracle and/or its affiliates.

  - Add and configure any new keys that you want to use.
  - Delete the keys that were removed.
  - If a key's default value changed, determine whether you want to modify the key's value.
- Close and save the files as *updatedOldOverrideValues*.

4. Create an **upgrade-billing.yaml** file and set these Billing Care and Billing Care REST API keys:
   - **ocbc.bc.deployment.app.imageTag**: Set this to the new release number in the format 16.26.0.*x.x.x* for patch sets and 16.26.0.*x.x.x-nnnnnnn* for interim patches.
   - **ocbc.bcws.deployment.app.imageTag**: Set this to the new release number in the format 16.26.0.*x.x.x* for patch sets and 16.26.0.*x.x.x-nnnnnnn* for interim patches.

   The **upgrade-billing.yaml** file will be used with both **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.
5. Deploy the 16.26.0 versions of Billing Care and Billing Care REST API with the latest changes by running the Helm upgrade command for **oc-cn-op-job-helm-chart**:

   ```
   helm upgrade oldOpJobReleaseName oc-cn-op-job-helm-chart --values
   updatedOldOverrideValues --values upgrade-billing.yaml --namespace oldBrmNameSpace
   ```

   Wait for the jobs to complete their tasks.
6. Start your 16.26.0 versions of the Billing Care and Billing Care REST API services by running the Helm upgrade command for **oc-cn-helm-chart**:

   ```
   helm upgrade oldBRMReleaseName oc-cn-helm-chart --values updatedOldOverrideValues --
   values upgrade-billing.yaml --namespace oldBrmNamespace
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

18

Performing Zero-Downtime Upgrades

Learn how to upgrade an Oracle Communications Billing and Revenue Management (BRM) cloud native deployment without having to take the environment offline. A zero-downtime upgrade allows your customers to continue using BRM's major services during the upgrade process.

Topics in this document:

- Performing a Zero-Downtime Upgrade of BRM
- Performing a Zero Downtime Upgrade of PDC

In this document, the BRM release running on your production system is called the *existing* release. The release you are upgrading to is called the *new* release. For example, if you upgrade from BRM 15.*x* to BRM 16.26.0, 15.*x* is the existing release and 16.26.0 is the new one.

Performing a Zero-Downtime Upgrade of BRM

You can perform a zero-downtime upgrade of your BRM cloud native services and the BRM database schema from 15.*x* to 16.26.0.

To perform a zero-downtime upgrade of BRM cloud native:

1. Download the BRM 16.26.0 cloud native package from the Oracle Software Delivery website (https://support.oracle.com) or the Oracle Support website (https://support.oracle.com).
2. Perform the prerequisite steps in the KM KB928854.
3. Prepare your existing BRM 15.*x* deployment for the zero-downtime upgrade:
   a. In your **override-values.yaml** file for the BRM 15.*x* **oc-cn-helm-chart**, set the following keys:

      ```
      ocbrm:
         refreshInterval: 10
         terminationGracePeriodSeconds: 120
         pcpReconnectDelayOnSocketError: 10
         pcpConnectRetryDelayOnError: 10
      ```

   b. In **oc-cn-helm-chart/templates/configmap_pin_conf_cm.yaml**, set the following entry:

      ```
      - cm pcm_connect_max_retries 10
      ```

   c. Ensure that at least two replica pods of cm, dm-oracle, dm-ifw-sync, and realtime-pipeline are running.
   d. Run the **helm upgrade** command for the BRM 15.*x* **oc-cn-helm-chart**:

      ```
      helm upgrade Brm_15_ReleaseName oc-cn-helm-chart --values OverrideValuesFile --
      namespace Brm_15_NameSpace
      ```

      where:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- *Brm_15_ReleaseName* is the release name assigned to your existing 15.*x* **oc-cn-helm-chart** installation.
- *OverrideValuesFile* is the file name and path to the file that overrides the **oc-cn-helm-chart/values.yaml** file.
- *Brm_15_NameSpace* is the namespace for your existing 15.*x* BRM deployment.

e. Back up your existing BRM 15.*x* Helm charts.

f. Copy the BRM 16.26.0 **oc-init-db-helm-chart** and **oc-cn-helm-chart** Helm charts to your BRM cloud native environment.

4. Upgrade the BRM cloud native database schema to 16.26.0:

a. Ensure that the **ConfigCacheRefreshInterval** business parameter in **bus_params_system.xml** is set to **0**.

For information about how to set business parameters, see "Running Load Utilities through Configurator Jobs" in *BRM Cloud Native System Administrator's Guide*.

b. Upgrade the cloud native database to 16.26.0 using the **helm install** command:

```
helm install NewInitDbReleaseName oc-cn-init-db-helm-chart --values
OverrideValuesFile --namespace NewInitDbNameSpace
```

where:

- *NewInitDbReleaseName* is the release name assigned to your new 16.26.0 **oc-cn-init-db-helm-chart** installation.
- *NewInitDbNameSpace* is the namespace for your new 16.26.0 **oc-cn-init-db-helm-chart** deployment.

Verify that the database has upgraded successfully.

5. Upgrade the BRM cloud native server to 16.26.0 by doing the following:

> **Note**
>
> To run billing during the BRM cloud native server upgrade for non-production systems, set the next billing date in the **oc-cn-helm-chart/templates/configmap_env_common.yaml** file to the following:
>
> ```
> VIRTUAL_TIME_SETTING: "-m 2 billingDate"
> ```
>
> For example, to set the next billing date to Feb 12, 2025, set *billingDate* to 021210002025.

a. Ensure the **ConfigCacheRefreshInterval** business parameter in **bus_params_system.xml** is set to **0**.

For information about how to set business parameters, see "Running Load Utilities through Configurator Jobs" in *BRM Cloud Native System Administrator's Guide*.

b. In your **override-values.yaml** file for the 16.26.0 **oc-cn-helm-chart**, set the following keys:

```
ocbrm:
   refreshInterval: 10
   terminationGracePeriodSeconds: 120
   pcpReconnectDelayOnSocketError: 10
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
pcpConnectRetryDelayOnError: 10
virtual_time:
   enabled: true
   sync_pvt_time: 5
```

If TLS is enabled in your BRM deployment, also configure the TLS settings and wallet sources required by BRM 16.26.0. Set **ocbrm.sslMode** to **ONE_WAY** or **TWO_WAY**, as required for your deployment.

If you use external Kubernetes secrets for TLS wallets, create any required secrets in *Brm_15_NameSpace* before you run the Helm upgrade, and set the appropriate component-specific **extUserSSLWalletSecret** keys in **override-values.yaml**. You can use **ocbrm.extCustomSSLWalletSecret** as the shared TLS wallet source when you do not configure a component-specific wallet secret. See "Managing Wallet and KeyStore Certificates".

c. In the 16.26.0 **oc-cn-helm-chart/templates/configmap_pin_conf_cm.yaml** file, set the following:

```
- cm pcm_connect_max_retries 10
- cm pcm_em_proto_vers 0
```

d. In the 16.26.0 **oc-cn-helm-chart/templates/configmap_pin_conf_rtp_pipeline.yaml** file, set the following:

```
- cm-em pcm_em_proto_vers 0
```

e. In the 16.26.0 **oc-cn-helm-chart/templates/cm.yaml** file, set the following:

```
spec.template.spec.containers(- name: cm).livenessProbe.initialDelaySeconds: 60
```

f. In the 16.26.0 **oc-cn-helm-chart/templates/configmap_pin_conf_dm_oracle.yaml** file, set the following:

```
- dm dm_ignore_fld_mismatch_err 1
```

g. Upgrade the BRM cloud native server to 16.26.0 using the **helm upgrade** command:

```
helm upgrade Brm_15_ReleaseName oc-cn-helm-chart --values OverrideValuesFile --
namespace Brm_15_NameSpace
```

Verify that all 15.*x* pods terminate and all 16.26.0 pods come up and run with 16.26.0 images.

6. Revert the configuration values in your **override-values.yaml** file for the 16.26.0 **oc-cn-helm-chart**:

a. In non-production systems, set the **VIRTUAL_TIME_SETTING** parameter to the default value in your **oc-cn-helm-chart/templates/configmap_env_common.yaml** file:

```
VIRTUAL_TIME_SETTING: "-m 0"
```

b. In your **oc-cn-helm-chart/templates/configmap_pin_conf_rtp_pipeline.yaml** file, remove or comment out the following entry:

```
- cm-em pcm_em_proto_vers 0
```

c. In your **oc-cn-helm-chart/templates/configmap_pin_conf_cm.yaml** file, remove or comment out the following entry:

```
- cm pcm_em_proto_vers 0
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

d. In your **override-values.yaml** file for **oc-cn-helm-chart**, set the **ocbrm.refreshInterval** key to its original value.

e. Set the **ConfigCacheRefreshInterval** business parameter in **bus_params_system.xml** to its original value.

   For information about how to set business parameters, see "Running Load Utilities through Configurator Jobs" in *BRM Cloud Native System Administrator's Guide*.

f. Run the **helm upgrade** command for the 16.26.0 **oc-cn-helm-chart**:

```
helm upgrade Brm_15_ReleaseName oc-cn-helm-chart --values OverrideValuesFile --
namespace Brm_15_NameSpace
```

Performing a Zero Downtime Upgrade of PDC

You can perform a zero downtime upgrade of your PDC cloud native services and the PDC database schema from the 12.0 or 12.0 Patch Set release to the 16.26.0 release.

You do so using a two-namespace approach in which you create an instance of PDC cloud native in a standby namespace, redirect PDC traffic to services in the standby namespace, upgrade PDC cloud native to release 16.26.0 in your original namespace, and then redirect PDC traffic back to your original namespace.

To upgrade PDC in zero downtime upgrade mode:

1. Create a temporary namespace, such as **BrmStandbyNameSpace**.
2. Clone your PDC *OverrideValuesFile* file to *StandbyOverrideValuesFile*.
3. In your *StandbyOverrideValuesFile* file for **oc-cn-op-job-helm-chart**, set the following keys:

```
ocpdc:
   configEnv:
      rcuPrefix: NewPrefix
      crossRefSchemaUserName: XrefSchema
      pdcSchemaUserName: PdcSchema
      siteNo: 2
      upgrade: true
```

   where:

   - *NewPrefix* is the new prefix for the PDC domain RCU schema.
   - *XrefSchema* is the same XREF schema name used for deploying PDC in **BrmNameSpace**.
   - *PdcSchema* is the same PDC schema name used for deploying PDC in **BrmNameSpace**.

4. In your *StandbyOverrideValuesFile* file for **oc-cn-helm-chart**, set the following keys:

```
ocpdc:
   configEnv:
      upgrade: true
```

   These settings will upgrade the PDC and XREF schema.

5. Copy the following templates from your **BrmNameSpace** Helm chart **template** directory to the **BrmStandbyNameSpace** Helm chart **template** directory:

   - **secret_env_brm.yaml**
   - **configmap_pin_conf_brm_apps_2.yaml**

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **configmap_loadifwconfig_reg.yaml**
- **configmap_env_common.yaml**
- **virtual_time_pvc.yaml**
- **configmap_infranet_properties_brm_apps.yaml**
- **config_jobs.yaml**
- **storage_class_green.yaml**
- **realtime_pipeline_common_pvc.yaml**
- **configmap_tns_admin.yaml**
- **secret_wallet_db.yaml**
- **_helpers_utils.tpl**
- **configmap_pdc_aux_engines.yaml**
- **configmap_log_properties_pdc.yaml**
- **configmap_env_pdc.yaml**
- **configmap_env_pdc_rre.yaml**
- **job_ie_pdc.yaml**
- **domain_pdc.yaml**
- **_pdchelpers.tpl**
- **pdc_domain_monitoring_role.yaml**
- **pdc_domain_monitoring_rbac.yaml**
- **service_monitor_pdc_domain.yaml**
- **secret_pdc.yaml**
- **volume_pdc_brm.yaml**
- **deployment_pdc_rre.yaml**
- **deployment_pdc_bre.yaml**
- **deployment_pdc_syncpdc.yaml**

6. Create a configuration file named **cm-service-external-name.yaml** and add the following content:

```
apiVersion: v1
kind: Service
metadata:
  name: cm
  namespace: BrmStandbyNameSpace
spec:
  externalName: cm.BrmNameSpace.svc.cluster.local
  internalTrafficPolicy: Cluster
  ports:
  - port: 11960
    protocol: TCP
    targetPort: 11960
  sessionAffinity: None
  type: ExternalName
```

7. Apply the configuration file to a resource:

```
kubectl apply -f cm-service-external-name.yaml
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

8. Deploy the PDC 12.0 Patch Set 7 or 8 Helm charts in your standby namespace:

```
helm install OpJobStandbyReleaseName oc-cn-op-job-helm-chart --values
StandbyOverrideValuesFile --namespace BrmStandbyNameSpace

helm install BrmStandbyReleaseName oc-cn-helm-chart --values
StandbyOverrideValuesFile --namespace BrmStandbyNameSpace
```

9. Redirect PDC traffic to services in **BrmStandbyNameSpace**.
10. Upgrade your PDC cloud native services to release 16.26.0 in **BrmNameSpace** while requests are temporarily routed to **BrmStandbyNameSpace**.
    a. Set the following keys in your **override-values.yaml** file for **oc-cn-op-job-helm-chart**:

```
ocpdc:
   configEnv:
      siteNo: 1
      upgrade: true
```

    b. Follow the instructions in "Upgrading Your PDC Cloud Native Services" to upgrade your original namespace to 16.26.0.
11. Redirect PDC traffic back to services in your original namespace (**BrmNameSpace**).

Installing PDC 15.*x* on More Than Two Sites

You can install PDC on Sites 1 and 2 with the default configuration and start using them immediately.

When you deploy PDC across more than two sites with the same PDC and cross-reference schema, such as during a zero-downtime upgrade of disaster recovery systems, you must follow the steps below on Site 3 and all subsequent sites.

> **Note**
>
> You can restore PDC using **oc-cn-helm-chart** if there are any issues with Sites 1 or 2.

To deploy PDC on Site 3 and subsequent sites:

1. Create a Python script on the machine where you are installing the third site named **sample_unique_WLSTORE.py** with content similar to the following:

```
print("\n[INFO] Start - Custom Weblogic WLST - To configure unique
WLSTORE  ...")

host = os.getenv("HOSTNAME", "localhost")
connect("WLSAdminUser", "password", "t3://" + host + ":WLSPort")

edit()
startEdit()

cd("/JDBCStores/JDPersistentJDBCStore")
cmo.setDataSource(getMBean("/JDBCSystemResources/
JobDispatcherPersistentDS"))
cmo.setPrefixName("PDCJD3_")
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
save()
activate()
disconnect()

print("\n[INFO] Completed - Custom Weblogic WLST - configured unique
WLSTORE ...")
```

2. In the "connect" line of the file, replace the following:
   - *WLSAdminUser* is the user name of the administrative user for WebLogic Server.
   - *password* is the password of your WebLogic Server.
   - *WLSPort* is the listener port for the WebLogic administration server.
3. Save the **Sample_unique_WLSTORE.py** file to the **oc-cn-op-job/pdc/customWLSPython/** directory.
4. Set the file permissions for **Sample_unique_WLSTORE.py** to **755**.
5. In your **override-values.yaml** file for **oc-cn-op-job-helm-chart**, set the **isCustomWLSPython** key to **true**.
6. Create WebLogic domains by running the 15.*x* version of **oc-cn-op-job-helm-chart** from the **helmcharts** directory:

   `helm install` *`OpJobReleaseName`* **`oc-cn-op-job-helm-chart --namespace`** *`BrmNameSpace`* **`--values override-values.yaml`**

Copyright © 2019, 2026, Oracle and/or its affiliates.

19

Performing Zero-Downtime Upgrades of Disaster Recovery Cloud Native Systems

Learn how to perform a zero-downtime upgrade of an Oracle Communications Billing and Revenue Management (BRM) cloud native deployment in an active-active disaster recovery system.

Topics in this document:

- About the Zero-Downtime Upgrade of an Active-Active Disaster Recovery System
- Tasks for Upgrading a BRM Cloud Native Active-Active System

About the Zero-Downtime Upgrade of an Active-Active Disaster Recovery System

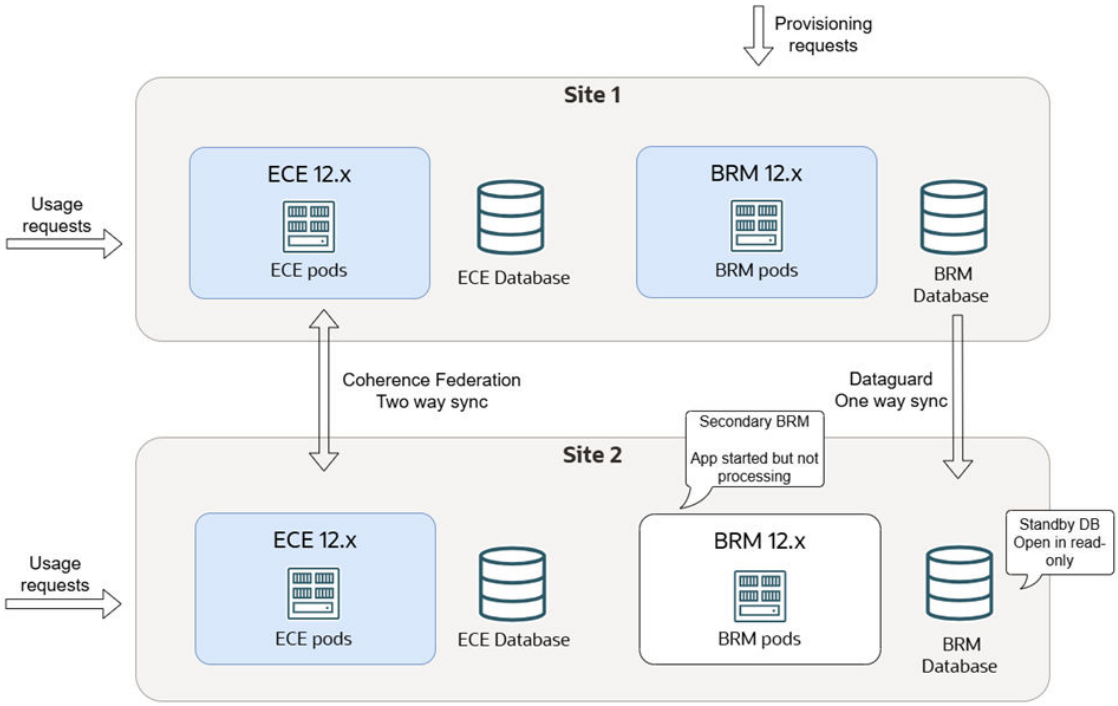
The steps for performing a zero-downtime cloud native upgrade of an active-active disaster recovery system assume that you are upgrading to the new release and assumes that your system contains a secondary BRM and ECE instance, which can be on a disaster recovery site or the primary site.

> ⚠ **Caution**
>
> Be aware that during the upgrade process:
>
> - Usage processing is done in a single instance only.
> - PDC is not available.
> - ECE cache federation is supported from the old to the new version but not from the new to the old version.
> - If an Oracle Data Guard role reversal is required for the BRM database, a short (few minutes) downtime occurs for provisioning.

Figure 19-1 shows initial state of a BRM cloud native active-active disaster recovery system.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 19-1 Initial State of BRM Cloud Native Active-Active Disaster Recovery System



For information about the supported software versions, see *BRM Compatibility Matrix*.

Tasks for Upgrading a BRM Cloud Native Active-Active System

To upgrade your BRM cloud native active-active disaster recovery system using the zero-downtime upgrade process:

1. Turn off Site 2. See "Switching Off Site 2".
2. From Site 2, uninstall the old version of BRM and ECE. See "Uninstalling BRM and ECE from Site 2".
3. On Site 2, upgrade the Kubernetes platform and all prerequisite software to the versions required by the 15.*x* release. See *BRM Compatibility Matrix* for the list of supported software versions.

   For information, see "Upgrade a Cluster" in the Kubernetes documentation.
4. On Site 2, upgrade the BRM database schema to the 15.*x* release. See "Upgrading Your BRM Database Schema in Site 2".
5. On Site 2, install the BRM 15.*x* cloud native software. See "Installing BRM 15.*x* Cloud Native on Site 2".
6. From Site 2, drop the old persistence database schema. See "Dropping the ECE Persistence Database Schema from Site 2".
7. On Site 2, install ECE 15.*x*. See "Installing ECE 15.*x* Cloud Native on Site 2".
8. Transfer usage processing from Site 1 to Site 2. See "Failing Over Site 1 to Site 2".
9. From Site 1, uninstall the old version of BRM and ECE. See "Uninstalling BRM and ECE from Site 1".

Copyright © 2019, 2026, Oracle and/or its affiliates.

10. On Site 1, upgrade the Kubernetes platform and all prerequisite software to the versions required by the 15.*x* release. See *BRM Compatibility Matrix* for the list of supported software versions.

    For information, see "Upgrade a Cluster" in the Kubernetes documentation.

11. From Site 1, drop the old ECE persistence database schema. See "Dropping the ECE Persistence Database Schema from Site 1".

12. On Site 1, install ECE 15.*x*. See "Installing ECE 15.*x* Cloud Native on Site 1".

13. Restart the federation process between Site 1 and Site 2. See "Federating ECE Cache Data Between Site 1 and Site 2".

14. If required, move the provisioning flow back to Site 1 and do an Oracle Data Guard role reversal.

> **Note**
>
> The Data Guard role reversal may cause a few minutes of downtime in the provisioning flow, but it does not impact the usage flow.

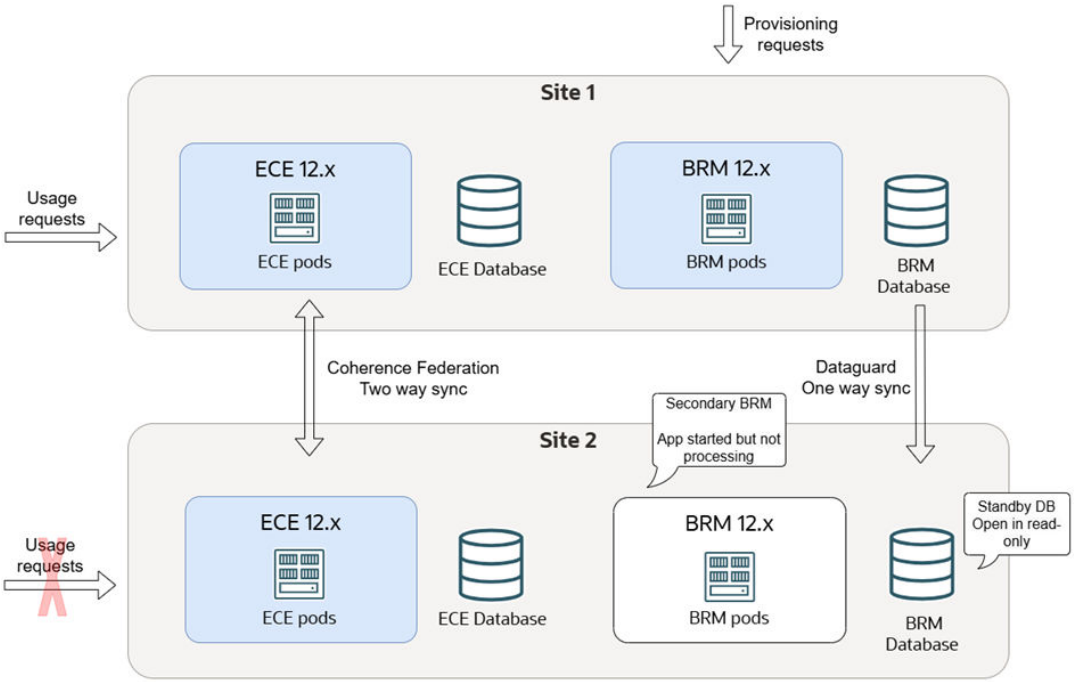
Switching Off Site 2

During the first phase of the upgrade process, Site 1 processes usage requests while you upgrade the software in Site 2.

To switch off Site 2:

1. Stop all usage requests to Site 2 and direct all usage requests to Site 1, as shown in Figure 19-2.

Figure 19-2 Usage Requests for Switching Off Site 2



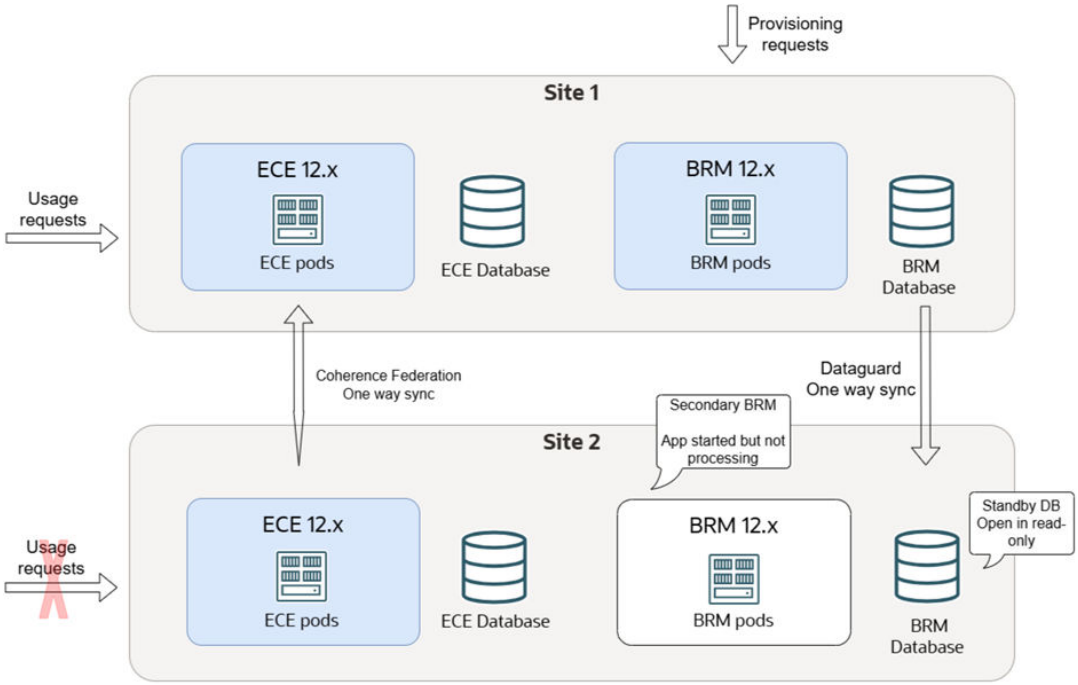
Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Stop the connection from BRM on Site 1 to ECE on Site 2. To do so:
   - On ECE Site 2, stop the EM Gateway.
   - On BRM Site 1, remove any connection to the EM Gateway on Site 2.
3. On ECE Site 1, mark Site 2 as failed.

   This stops ECE from forwarding rating requests to Site 2 for subscribers with Site 2 as their preferred site. Usage requests are now rated on Site 1.
4. Disable the federation from ECE Site 1 to Site 2, as shown in Figure 19-3.

> **Note**
>
> Keep the Site 2 to Site 1 federation active to drain any remaining data from Site 2.

Figure 19-3 Disabled Federation from ECE Site 1 to Site 2



5. Check that the federation backlog from Site 2 to Site 1 has been cleared.

   All traffic to Site 2 is stopped. Ensure all data from Site 2 has been synchronized with Site 1. The Coherence Federation Metrics should show as IDLE instead of YIELDING.
6. On Site 2, check that all rated events in the ECE cache have been extracted by running the following with the **query.sh** utility:

```
./query.sh
Coherence Command Line Tool
CohQl> select value() from AggregateObjectUsage
```

   If successful, the query displays zero entries. See "query" in *ECE Implementing Charging* for information about the utility's syntax and parameters.

Copyright © 2019, 2026, Oracle and/or its affiliates.

7. On Site 2, use SQL*Plus to check that all Site 2 rated events have been extracted from the persistence database and are present in BRM:

```
sqlplus pin@databaseName
Enter password: password

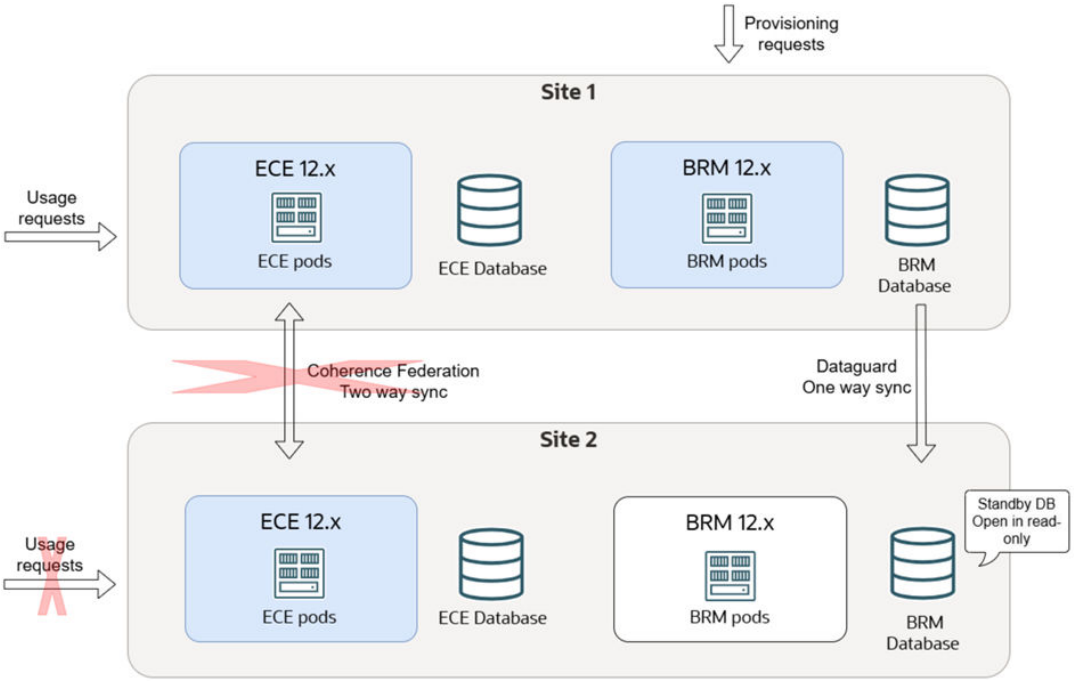
SQL> select count(*) from ratedevent_site2Name
```

where *databaseName* is the service name or database alias of the BRM database, and *password* is the password for the **pin** user.

> **Note**
>
> The Site 2 persistence database might contain some Site 1 events. After the Site 2 to Site 1 federation process is stopped, these events are not extracted or purged. However, they will be processed on Site 1. You can ignore these events because they get purged when you re-create the Site 2 persistence database later.

8. Stop the federation process from Site 2 to Site 1, as shown in Figure 19-4.

Figure 19-4 Stopped Federation from ECE Site 2 to Site 1



All data is now synchronized with Site 1. Site 2 is isolated and ready for the upgrade.

Uninstalling BRM and ECE from Site 2

To uninstall the old version of BRM and ECE from Site 2:

1. Uninstall the old version of BRM cloud native from Site 2:

```
helm uninstall BrmReleaseName --namespace BrmNameSpace
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Uninstall the old version of ECE cloud native from Site 2:

   ```
   helm uninstall EceReleaseName --namespace BrmNameSpace
   ```

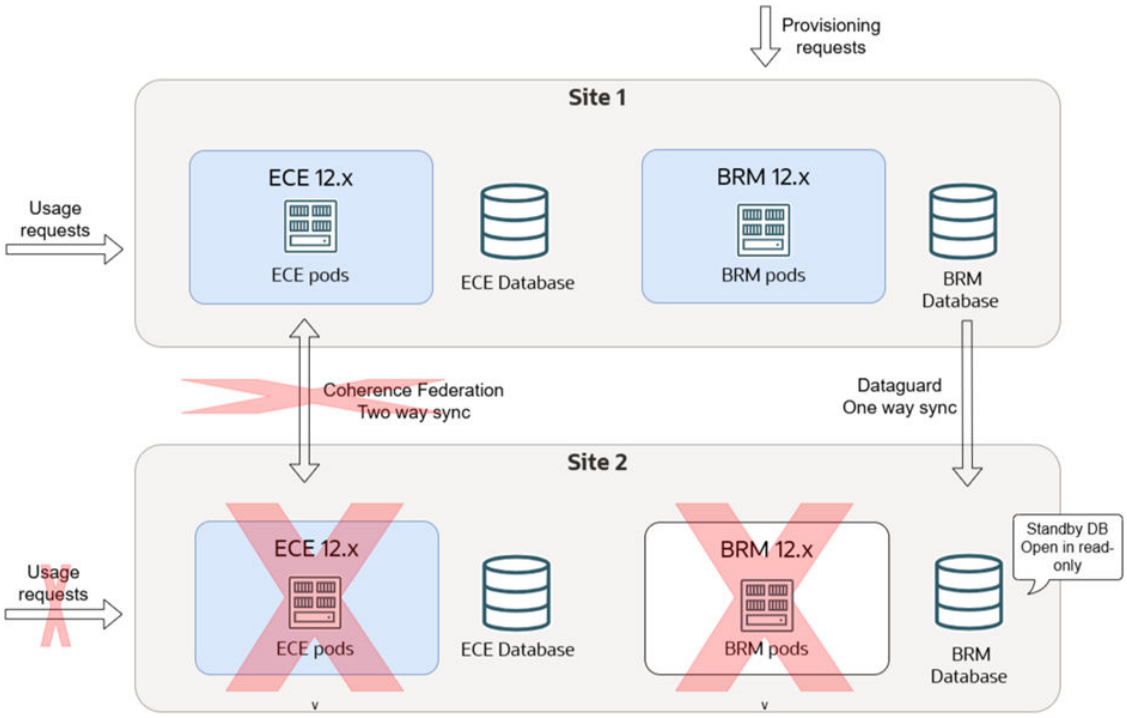
   where:

   - *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.
   - *EceReleaseName* is the release name for **oc-cn-ece-helm-chart** and is used to track this installation instance.
   - *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.

These commands delete all the resources associated with the chart's last release and the release history.

Figure 19-5 shows how to uninstall BRM and ECE from Site 2.

Figure 19-5 Uninstall BRM and ECE from Site 2



Upgrading Your BRM Database Schema in Site 2

To upgrade your BRM database schema in Site 2 to release BRM 15.*x*:

1. Download and extract the BRM 15.*x* cloud native database initializer Helm chart (**oc-cn-init-db-helm-chart**) from Oracle Software Delivery Cloud (https://edelivery.oracle.com).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files" for more information.

Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Copy your old wallet files (**ewallet.p12** and **cwallet.sso**) from the dm-oracle pod's **/oms/wallet/client** directory to the BRM 15.*x* Helm chart's **oc-cn-init-db-helm-chart/existing_wallet/** directory.

   > **Note**
   >
   > Copying old wallet files is applicable only if the base version is 12.*x*.

3. Create an **override-init-db-15.yaml** file for the 15.*x* version of **oc-cn-init-db-helm-chart**.
4. In your **override-init-db-15.yaml** file, set the following keys:
   - **ocbrm.is_upgrade**: Set this to **true**.
   - **ocbrm.wallet.client**: Set this to the password for the client wallet. This value must match that of your old release.
   - **ocbrm.wallet.server**: Set this to the password for the server wallet. This value must match that of your old release.
   - **ocbrm.wallet.root**: Set this to the password for the root wallet. This value must match that of your old release.
5. Enter this command from the **helmcharts** directory to upgrade the database schema. Ensure you run the 15.*x* **oc-cn-init-db-helm-chart** Helm chart with a new release name and namespace.

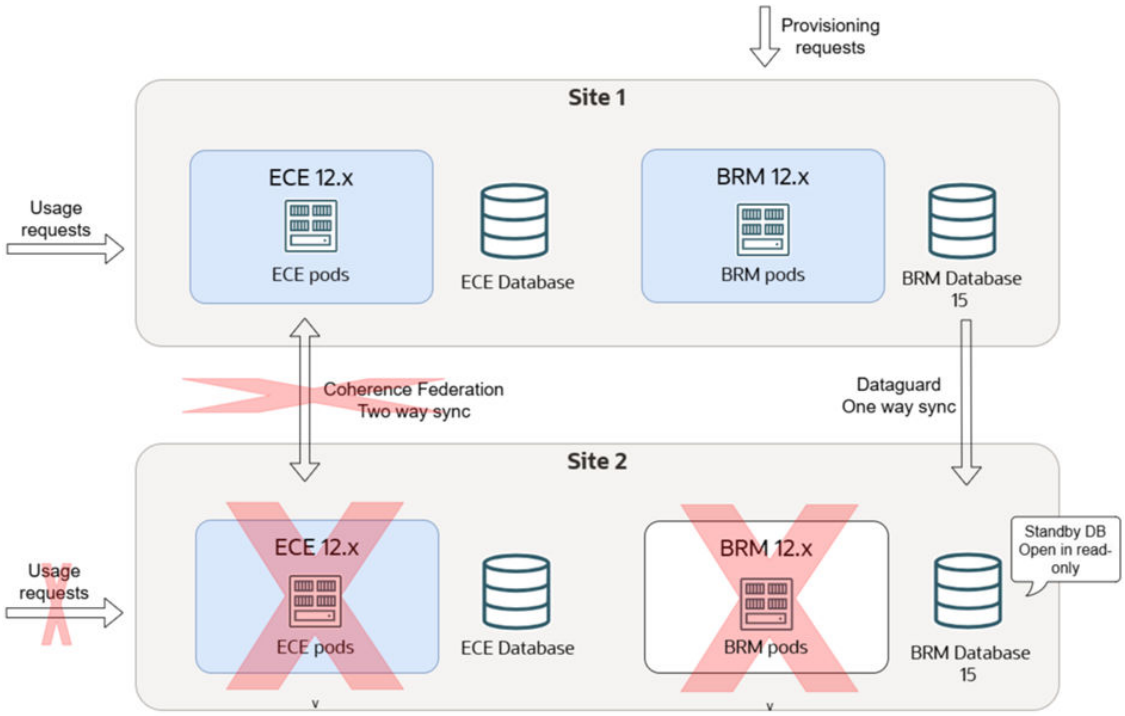
   ```
   helm install newInitDbReleaseName oc-cn-init-db-helm-chart --namespace
   newInitDbNameSpace --values oldOverrideValues --values override-init-
   db-15.yaml
   ```

   where:
   - *newInitDbReleaseName* is the new release name for the 15.*x* version of **oc-cn-init-db-helm-chart**.
   - *newInitDbNameSpace* is the new namespace for the 15.*x* version of **oc-cn-init-db-helm-chart**.
   - *oldOverrideValues* is the name and path to your old **override-values.yaml** file for **oc-cn-init-db-helm-cart**.

Figure 19-6 shows an upgraded BRM database schema in Site 2.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 19-6 Upgraded BRM Database Schema in Site 2



Installing BRM 15.*x* Cloud Native on Site 2

To configure and deploy BRM 15.*x* on your cloud native system on Site 2:

1. Download and extract the BRM 15.*x* cloud native (**oc-cn-helm-chart**) and BRM 15.*x* Operator Job Helm Chart (**oc-cn-op-job-helm-chart**) from Oracle Software Delivery Cloud (https://edelivery.oracle.com).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files" for more information.

2. To reuse your old SSL KeyStore with the new release, copy the PDC KeyStore files from the old Helm chart to the 15.*x* **oc-cn-op-job-helm-chart/pdc/pdc_keystore/** directory.

3. Create an **override-values-15.yaml** file.

   You will use this file with the 15.*x* version of **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.

4. In your **override-values-15.yaml** file, set the following keys:

   - In the BRM section:
     - **ocbrm.is_upgrade**: Set this to **true**.
     - **ocbrm.db.***: Set the BRM database schema details to the same values as your old release.
   - In the PDC section:
     - **ocpdc.configEnv.pdcSchemaUserName**: Set this to the same value as your old release.
     - **ocpdc.configEnv.crossRefSchemaUserName**: Set this to the same value as your old release.

Copyright © 2019, 2026, Oracle and/or its affiliates.

  - **ocpdc.configEnv.rcuPrefix**: Set this key to a new prefix to create a new RCU schema.
  - **ocpdc.configEnv.transformation.upgrade**: Set this to **true**. (For upgrades to 15.0.0 only)
  - **ocpdc.secretValue.walletPassword**: Set this to the same value as your old release.
  - **ocpdc.configEnv.deployAndUpgradeSite2**: Set this to **true**. (For upgrades to 15.0.1 on later)
  - **ocpdc.configEnv.upgrade**: Set this to **true**. (For upgrades to 15.0.1 or later)
- In the Business Operations Center section:
  - **ocboc.boc.configEnv.bocSchemaUserName**: Set this to the same value as your old release.
  - **ocboc.boc.configEnv.runUpgrade**: Set this to **true**.
  - **ocboc.boc.configEnv.rcuPrefix**: Set this to a new prefix to create a new RCU schema.
  - **ocboc.boc.secretVal.***: Set the Business Operations Center passwords to the same values as your old release.
- In the Billing Care section:
  - **ocbc.bc.configEnv.rcuPrefix**: Set this to a new prefix to create a new RCU schema.
  - **ocbc.bc.secretVal.***: Set the Billing Care passwords to the same values as your old release.
- In the Billing Care REST API section:
  - **ocbc.bcws.configEnv.rcuPrefix**: Set this to a new prefix to create a new RCU schema.
  - **ocbc.bcws.secretVal.***: Set the Billing Care REST API passwords to the same values as your old release.

5. Create WebLogic domains by running the 15.*x* version of **oc-cn-op-job-helm-chart** from the **helmcharts** directory:

   ```
   helm install oldOpJobReleaseName oc-cn-op-job-helm-chart --namespace
   oldBrmNameSpace --namespace oldOverrideValuesFile --values override-
   values-15.yaml
   ```

6. Install BRM cloud native services by running the 15.*x* version of **oc-cn-helm-chart** from the **helmcharts** directory:

   ```
   helm install oldBrmReleaseName oc-cn-helm-chart --namespace
   oldBrmNameSpace --values oldOverrideValuesFile --values override-
   values-15.yaml
   ```

Dropping the ECE Persistence Database Schema from Site 2

To drop the ECE persistence database schema from Site 2:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Run the following command:

   ```
   DROP USER schemaname CASCADE
   ```

2. Uninstall the old version of the ECE Helm chart:

   ```
   helm uninstall oldEceReleaseName
   ```

   where *oldEceReleaseName* is the release name for your old version of **oc-cn-ece-helm-chart**.

3. Delete the old version of **ece-persistence-job** from your system by running this command:

   ```
   kubectl delete job ece-persistence-job --namespace oldBrmNameSpace
   ```

   where *oldBrmNameSpace* is the namespace for your old version of your BRM Helm release.

> **Note**
>
> The ECE 15.*x* Helm chart re-creates the persistence database schema when you install it later.

Installing ECE 15.*x* Cloud Native on Site 2

To install ECE 15.*x* cloud native on Site 2:

1. Download and extract the ECE 15.*x* cloud native (**oc-cn-ece-helm-chart**) from Oracle Software Delivery Cloud (https://edelivery.oracle.com).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files" for more information.

2. Create an **override-values-ece-15.yaml** file.

3. In the file, configure your ECE 15.*x* cloud native services by following the instructions in "Configuring ECE Services".

4. Deploy the ECE 15.*x* cloud native services by entering this command from the **helmcharts** directory:

   ```
   helm install oldEceReleaseName oc-cn-ece-helm-chart --namespace
   oldBrmNameSpace --values oldOverrideValuesFile --values override-values-
   ece-15.yaml
   ```

   where:

   - *oldEceReleaseName* is the release name for your old version of **oc-cn-ece-helm-chart**.
   - *oldBrmNameSpace* is the namespace for your old **oc-cn-helm-chart** deployment.
   - *oldOverrideValuesFile* is the file name and path to the old version of your **override-values.yaml** file for **oc-cn-ece-helm-chart**.

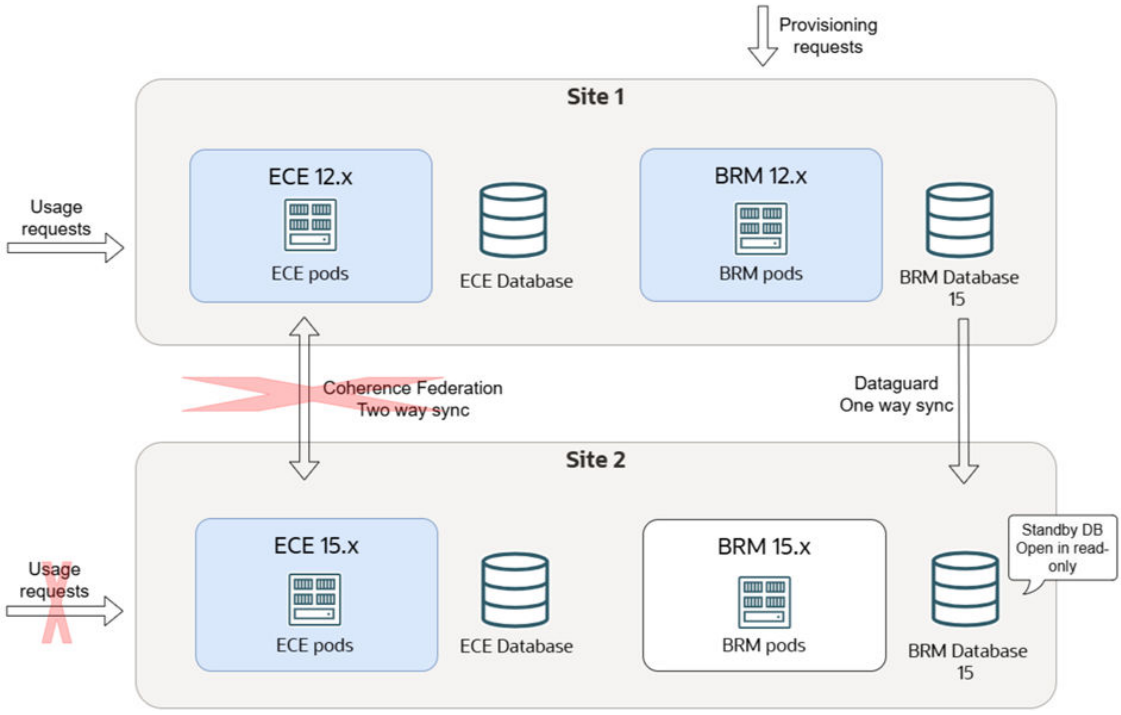
Copyright © 2019, 2026, Oracle and/or its affiliates.

Failing Over Site 1 to Site 2

Site 2 has been upgraded at this stage, but it is not handling any usage requests. You now switch usage request processing from Site 1 to Site 2.

Figure 19-7 shows failing over Site 1 to Site 2.

Figure 19-7 Failing Over Site 1 to Site 2



To fail over from Site 1 to Site 2:

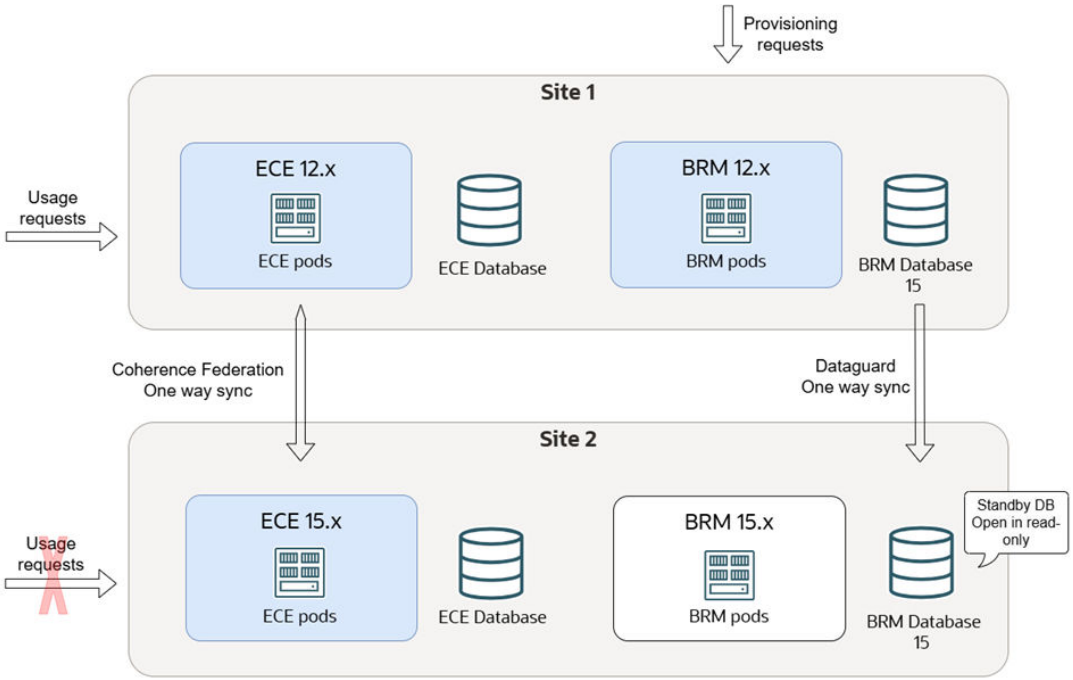
1. Delete the **NFProfileConfiguration** and **NFServiceConfiguration** ECE configuration MBeans from the site running the ECE 12.0 patch set (production site) before federating cache data to the backup site from the production site.

   See "Changing the ECE Configuration During Runtime" in *BRM Cloud Native System Administrator's Guide* for information about changing ECE configuration MBeans.

2. Start the Coherence federation process from Site 1 to Site 2, as shown in Figure 19-8. This provisions the empty ECE Site 2 cache with the latest data from Site 1.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 19-8 Coherence Federation Process from Site 1 to Site 2



3. On Site 2, mark Site 1 inactive to ensure no preferred site routing occurs from Site 2 to Site 1.

> **Note**
>
> Site 1 also has Site 2 marked as inactive.

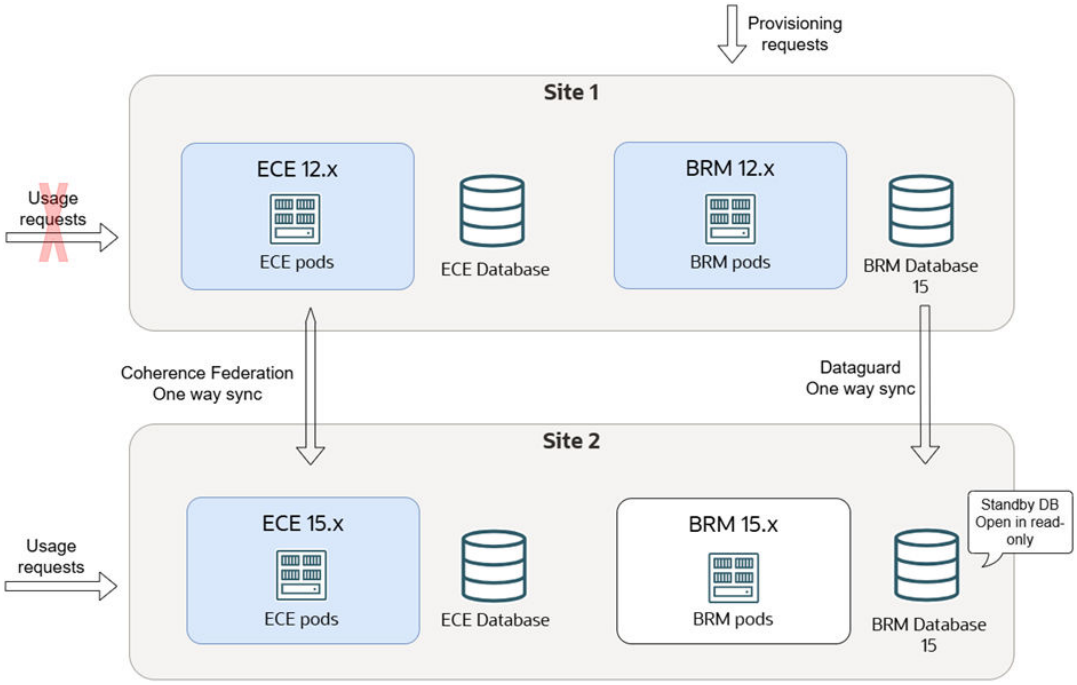
4. Check that the federation process is up to date. After the federation process completes successfully:
   - The ECE pods in Site 2 transition to the **Running** state.
   - Site 2 transitions to the **Usage Processing** state and spawns Monitoring Agent pods.

> **Note**
>
> Follow the steps in "Reloading ECE Application Configuration Changes" in the *BRM Cloud Native System Administrator's Guide* to update the values:
>
> - Under **brmGatewayConfigurations** set the **kafkaPartition** value to **3**.
> - Under **httpGatewayConfigurations** set the **kafkaPartition** value to **5,6,7,8,9**.

5. Stop all usage requests to Site 1 and then redirect them to Site 2, as shown in Figure 19-9.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 19-9 Usage Requests for Failing Over Site 1 to Site 2

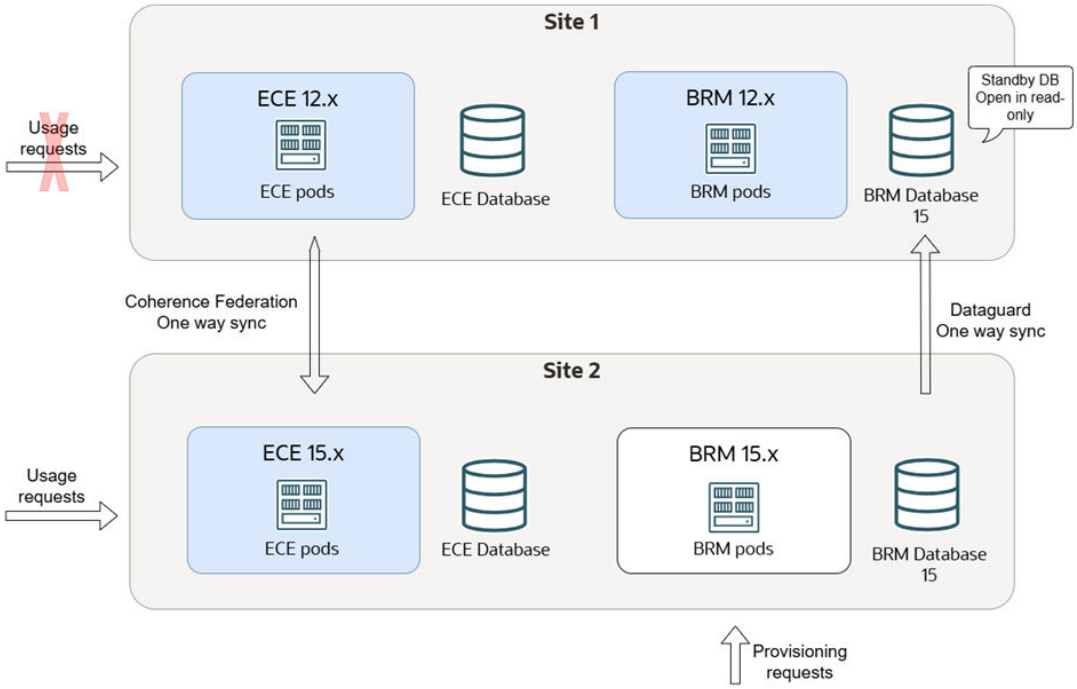


6. Switch provisioning to BRM on Site 2, as shown in Figure 19-10. To do so:
   a. On BRM Site 2, remove any connections to EM Gateway on Site 1.
   b. On the client side, switch provisioning to BRM on Site 2.
   c. On ECE Site 1, stop the EM Gateway.

> **Note**
>
> If latency between Site 1 and Site 2 is too high to give you acceptable performance for the provisioning flow, reverse the Oracle Data Guard roles and make the BRM database on Site 2 active. Be aware that this can cause a service interruption of a few minutes for provisioning.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 19-10 Switched Provisioning to BRM on Site 2



7. Check that the federation process from Site 1 to Site 2 has completed.

   Now that all traffic to Site 1 has stopped, ensure all data from Site 1 has been synchronized with Site 2. The Coherence Federation Metrics should display **IDLE** instead of **YIELDING**.

8. Check that all rated events in the ECE Site 1 cache have been extracted by running the **query** utility:

   ```
   ./query.sh
   Coherence Command Line Tool
   CohQl> select value() from AggregateObjectUsage
   ```

   If successful, this command returns zero entries.

9. On Site 1, check that all Site 1 rated events have been extracted from the persistence database and are present in BRM by using SQL*Plus:

   ```
   sqlplus pin@databaseName
   Enter password: password

   SQL> select count(*) from ratedevent_site1Name
   ```

   where *databaseName* is the service name or database alias of the BRM database, and *password* is the password for the **pin** user.

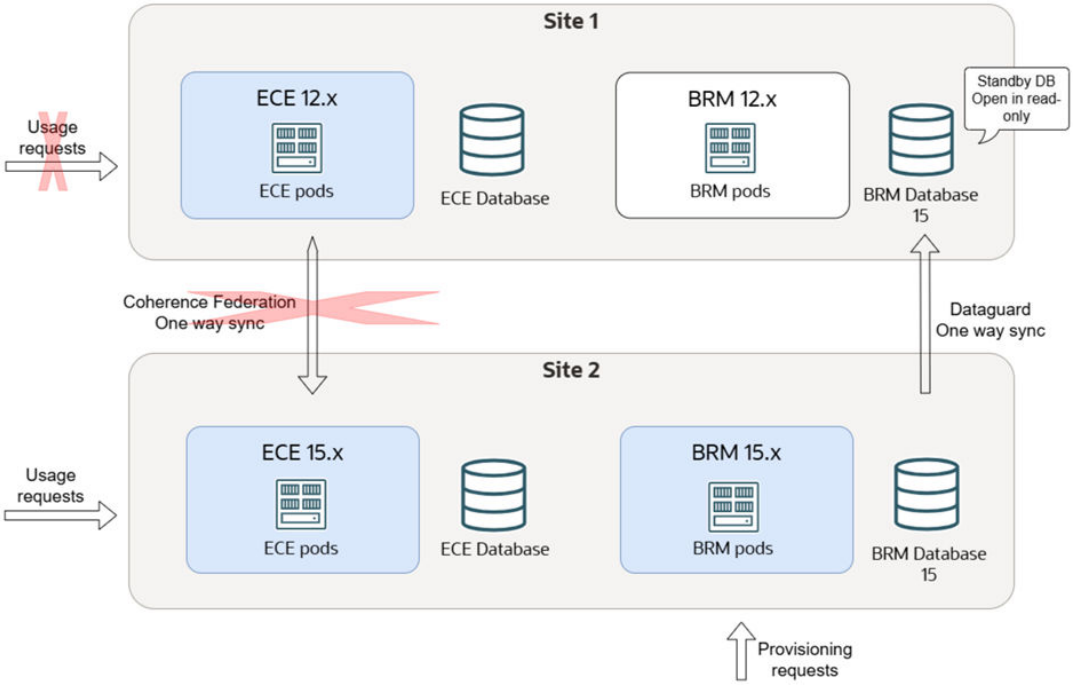
Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> The Site 1 persistence database might contain some Site 2 events. After the Site 2 to Site 1 federation process is stopped, these events are not extracted or purged. However, they will be processed on Site 2. You can ignore these events because they get purged when you re-create the Site 1 persistence database later.

10. Stop the federation process from Site 1 to Site 2, as shown in Figure 19-11.

    Site 1 is isolated and ready for the upgrade.

Figure 19-11 Stopped Federation Process from Site 1 to Site 2



Uninstalling BRM and ECE from Site 1

Uninstall the old version of BRM and ECE from Site 1.

To do so, perform the following steps on Site 1:

1. Uninstall the old version of BRM cloud native:

   ```
   helm uninstall BrmReleaseName --namespace BrmNameSpace
   ```

2. Uninstall the old version of ECE cloud native:

   ```
   helm uninstall EceReleaseName --namespace BrmNameSpace
   ```

where:

- *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance.

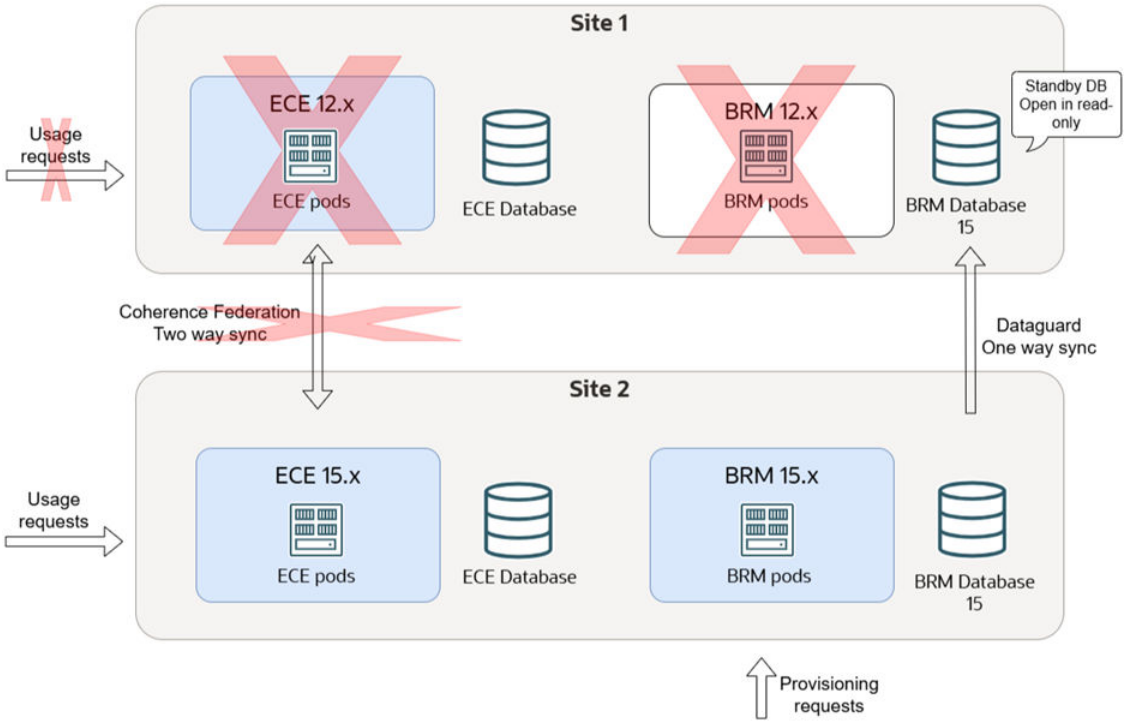
Copyright © 2019, 2026, Oracle and/or its affiliates.

- *EceReleaseName* is the release name for **oc-cn-ece-helm-chart** and is used to track this installation instance.
- *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM Helm chart.

These commands delete all of the resources associated with the chart's last release and the release history.

Figure 19-12 shows uninstallation of BRM and ECE from Site 1.

Figure 19-12 Uninstallation of BRM and ECE from Site 1



Installing BRM Cloud Native on Site 1

To configure and deploy BRM 15.*x* cloud native on Site 1:

1. Download and extract the BRM 15.*x* cloud native Helm chart (**oc-cn-helm-chart**) and BRM Operator Job Helm Chart (**oc-cn-op-job-helm-chart**) from Oracle Software Delivery Cloud (https://edelivery.oracle.com).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files" for more information.

2. To reuse your old SSL KeyStore, copy the PDC KeyStore files from the old Helm chart to the 15.*x* **oc-cn-op-job-helm-chart/pdc/pdc_keystore/** directory.

3. Create an **override-values-15.yaml** file.

   You will use this file with the 15.*x* version of **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.

4. In your **override-values-15.yaml** file, set the following keys:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- In the BRM section:
  - **ocbrm.is_upgrade**: Set this to **true**.
  - **ocbrm.db.***: Set the BRM database schema details to the same values as your old release.
- In the PDC section:
  - **ocpdc.configEnv.pdcSchemaUserName**: Set this to the same value as your old release.
  - **ocpdc.configEnv.crossRefSchemaUserName**: Set this to the same value as your old release.
  - **ocpdc.configEnv.rcuPrefix**: Set this key to a new prefix to create a new RCU schema. The value must be different from the one used in Site 2.
  - **ocpdc.configEnv.transformation.upgrade**: Set this to **true**. (For upgrades to 15.0.0 only)
  - **ocpdc.secretValue.walletPassword**: Set this to the same value as your old release.
  - **ocpdc.configEnv.deployAndUpgradeSite2**: Set this to **false**. (For upgrades to 15.0.1 or later)
  - **ocpdc.configEnv.upgrade**: Set this to **true**. (For upgrades to 15.0.1 or later)
- In the Business Operations Center section:
  - **ocboc.boc.configEnv.bocSchemaUserName**: Set this to the same value as your old release.
  - **ocboc.boc.configEnv.runUpgrade**: Set this to **true**.
  - **ocboc.boc.configEnv.rcuPrefix**: Set this to a new prefix for the RCU schema. The value must be different from the one used in Site 2.
  - **ocboc.boc.secretVal.***: Set the Business Operations Center passwords to the same values as your old release.
- In the Billing Care section:
  - **ocbc.bc.configEnv.rcuPrefix**: Set this to a new prefix for the RCU schema. This value must be different from the one used in Site 2.
  - **ocbc.bc.secretVal.***: Set the passwords to the same values as your old release.
- In the Billing Care REST API section:
  - **ocbc.bcws.configEnv.rcuPrefix**: Set this to a new prefix for the RCU schema. This value must be different from the one used in Site 2.
  - **ocbc.bcws.secretVal.***: Set the passwords to the same values as your old release.

5. Save and close the file.
6. Create WebLogic domains by running the 15.*x* version of **oc-cn-op-job-helm-chart** from the **helmcharts** directory:

   ```
   helm install oldOpJobReleaseName oc-cn-op-job-helm-chart --namespace
   oldBrmNameSpace --namespace oldOverrideValuesFile --values override-
   values-15.yaml
   ```

   where:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- *oldOpJobReleaseName* is the release name assigned to your old release of the **oc-cn-op-job-helm-chart** installation.
- *oldBrmNameSpace* is the namespace for your old version of the BRM deployment.
- *oldOverrideValuesFile* is the file name and path of your old version of the **override-values.yaml** file for **oc-cn-op-job-helm-chart**.

7. Install BRM cloud native services by running the 15.*x* version of **oc-cn-helm-chart** from the **helmcharts** directory:

   ```
   helm install oldBrmReleaseName oc-cn-helm-chart --namespace
   oldBrmNameSpace --values oldOverrideValuesFile --values override-
   values-15.yaml
   ```

   where *oldBrmReleaseName* is the release name assigned to your old version of the **oc-cn-helm-chart** installation.

Dropping the ECE Persistence Database Schema from Site 1

To drop the old persistence database schema from Site 1:

1. Run the following command:

   ```
   DROP USER schemaname CASCADE
   ```

2. Uninstall the old version of ECE Helm chart:

   ```
   helm uninstall oldEceReleaseName
   ```

   oldEceReleaseName is the release name for the old version of **oc-cn-ece-helm-chart**.

3. Uninstall the old version of **ece-persistence-job** from your system by running this command:

   ```
   kubectl delete job ece-persistence-job --namespace oldBrmNameSpace
   ```

   where *BrmNameSpace* is the namespace for the old version of the BRM Helm chart.

> **Note**
>
> The ECE Helm chart re-creates the database schema when you install it later.

Installing ECE 15.*x* Cloud Native on Site 1

To install ECE 15.*x* cloud native on Site 1:

1. Download and extract the ECE 15.*x* cloud native (**oc-cn-ece-helm-chart**) from Oracle Software Delivery Cloud (https://edelivery.oracle.com).

   See "Downloading Packages for the BRM Cloud Native Helm Charts and Docker Files" for more information.

2. Create an **override-values-ece-15.yaml** file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3. In the file, configure your ECE 15.*x* cloud native services by following the instructions in "Configuring ECE Services".
4. Deploy the ECE 15.*x* cloud native services by entering this command from the **helmcharts** directory:

   ```
   helm install oldEceReleaseName oc-cn-ece-helm-chart --namespace
   oldBrmNameSpace --values oldOverrideValuesFile --values override-values-
   ece-15.yaml
   ```

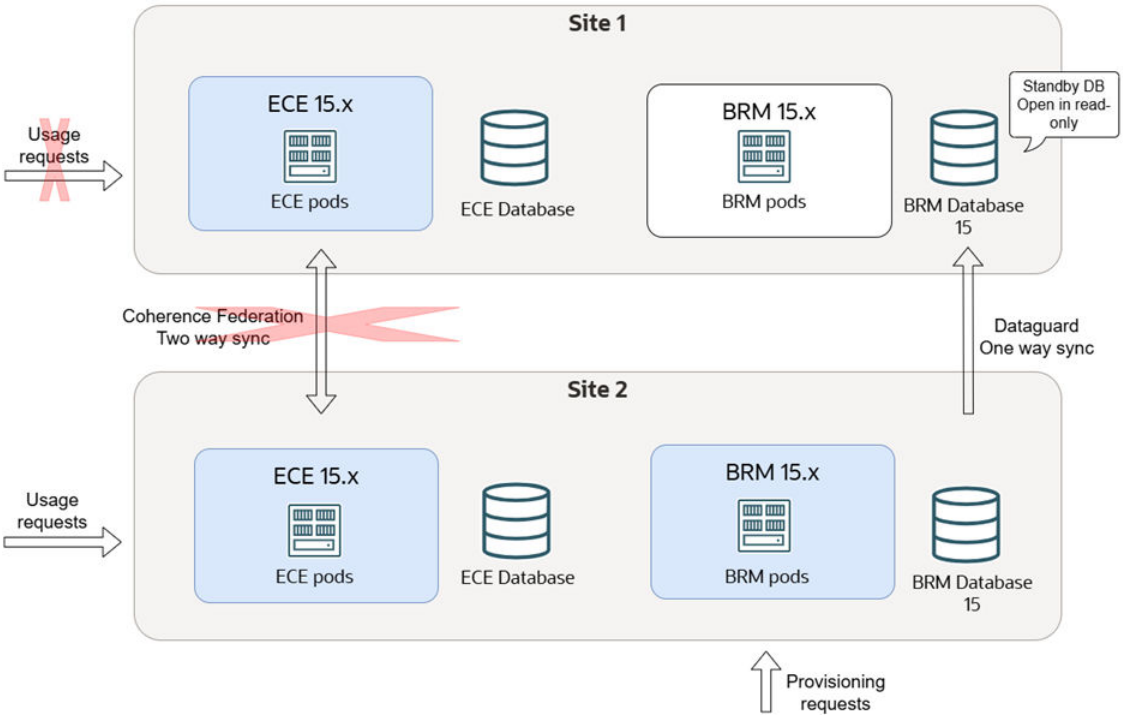
   where:

   - *oldEceReleaseName* is the release name for your old version of **oc-cn-ece-helm-chart**.
   - *oldBrmNameSpace* is the release name for your old version of **oc-cn-helm-chart** deployment.
   - *oldOverrideValuesFile* is the file name and path to the old version of your **override-values.yaml** file for **oc-cn-ece-helm-chart**.

Site 1 is now upgraded, but it is idle.

Figure 19-13 shows installation of ECE on Site 1.

Figure 19-13 Installation of ECE Cloud Native on Site 1



Federating ECE Cache Data Between Site 1 and Site 2

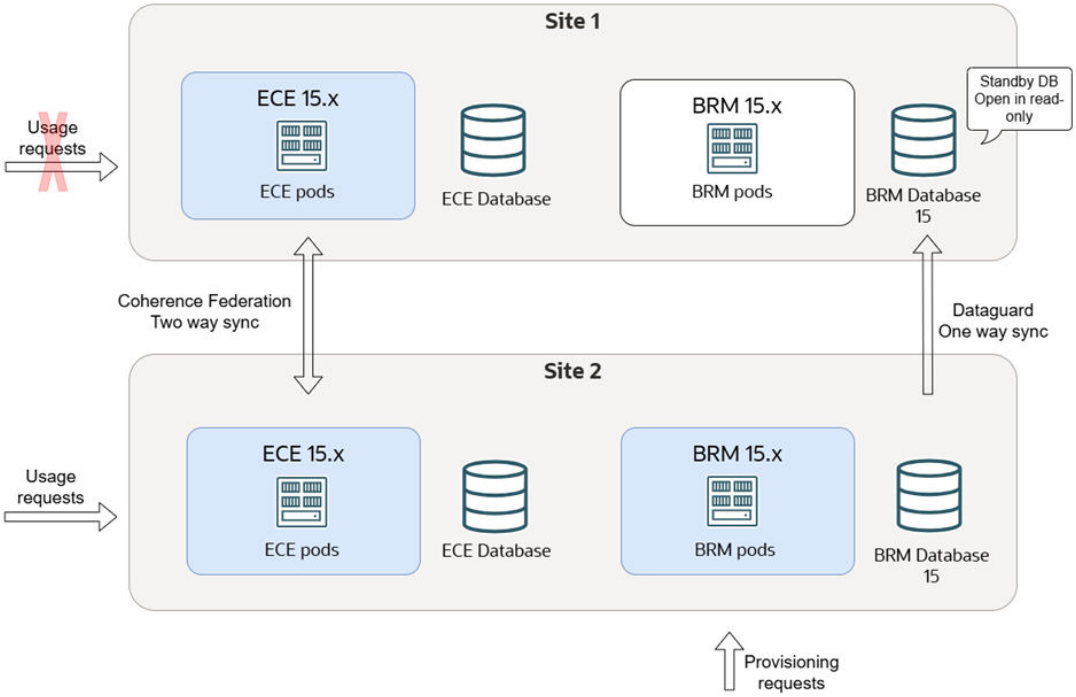
Now that Site 1 and Site 2 have been upgraded to release 15.*x*, you can restart the federation process between them.

To start the federation process of ECE cache data between Site 1 and Site 2:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Enable the two-way federation process between Site 1 and Site 2, as shown in Figure 19-14.

Figure 19-14 Federation of ECE Cache Data Between Sites



2. Check that the federation backlog is processed successfully.

   After the federation process completes, ECE in Site 1 transitions to the **Usage Processing** state and spawns the Monitoring Agent pods.

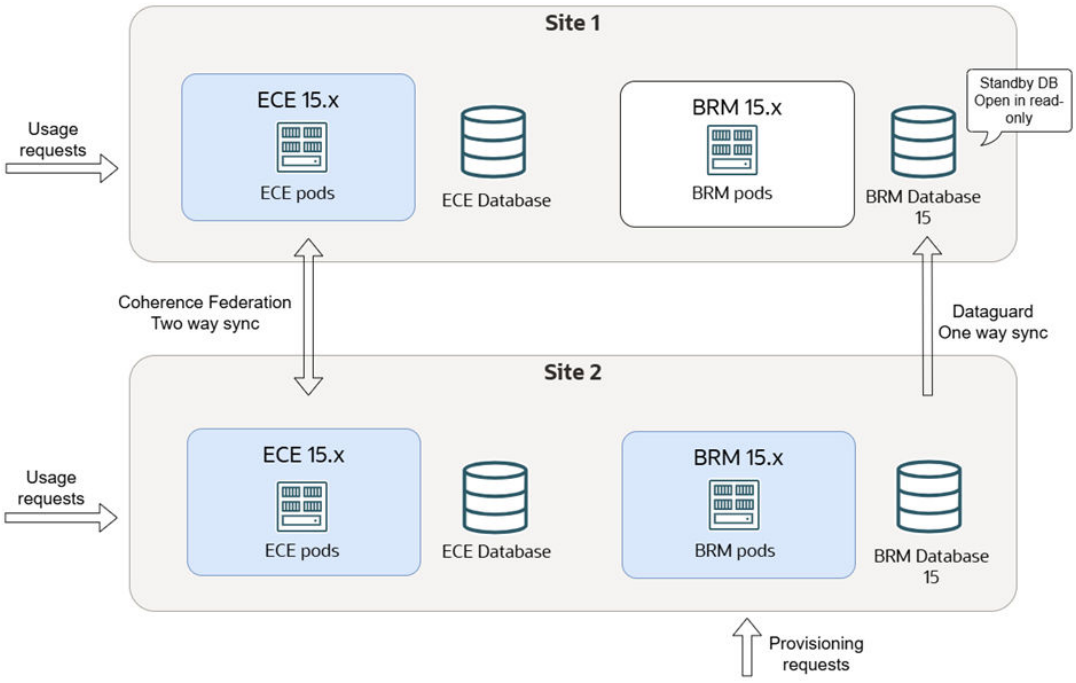
3. In ECE Site 1, mark ECE in Site 2 as active. Likewise, in ECE Site 2, mark ECE in Site 1 as active.

   This enables usage rating requests to be forwarded to the subscriber's preferred site.

4. Enable usage requests to Site 2, as shown in Figure 19-15.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 19-15 Enabled Usage Requests for Site 2



5. Enable failover connection to the EM Gateway. On both sides, ensure the CM has a failover EM connection to the EM Gateway on the other site.

Copyright © 2019, 2026, Oracle and/or its affiliates.

20

Migrating from On-Premise BRM to BRM Cloud Native

Learn how to migrate from an on-premise release of Oracle Communications Billing and Revenue Management (BRM) 7.5.*x* or 12.*x* to a BRM 16.26.0 cloud native release.

Topics in this document:

- Migrating to BRM Cloud Native
- Migrating from PDC On Premises to PDC Cloud Native

Migrating to BRM Cloud Native

To migrate from an on-premises release to BRM cloud native:

1. Upgrade your BRM and PDC database schemas to release 16.26.0:
   - If you are upgrading from a 12.0 or 12.0 Patch Set 1 database schema, follow the instructions in "Installing BRM 12.0 Patch Sets" in *BRM Patch Set Installation Guide*.
   - If you are upgrading from a 12.0 Patch Set 2 or later database schema, follow the instructions in "Upgrading Your Database Schema".

   > **Note**
   >
   > - To use an existing BRM database schema with your cloud native deployment, follow the instructions in "Deploying BRM with an Existing Schema".
   > - To use an existing Business Operations Center database schema, point to your existing Business Operations Center schema in the **override-values.yaml** file for **oc-cn-op-job-helm-chart**. See "Adding Business Operations Center Keys for oc-cn-op-job-helm-chart".

2. If you customized BRM or Billing Care, layer your customizations on top of the images provided with this release before deploying the images. For guidelines about customization, see "Customizing BRM Cloud Native Services".

Migrating from PDC On Premises to PDC Cloud Native

Before migrating your system, perform these steps:

- Ensure you have upgraded your PDC on-premises release to version 12.0 Patch Set 3 or later. Follow the instructions in "Upgrading Pricing Design Center Software".
- Back up your existing PDC and cross-reference database schemas.
- Download and load the PDC 16.26.0 cloud native images to the image repository or to the virtual machine (VM) where PDC is deployed.
- Download and extract the BRM 16.26.0 Helm charts.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Verify that BRM has been upgraded to the same 16.26.0 cloud native version as PDC cloud native.

The following are the high-level steps for migrating PDC from an on-premises release to cloud native:

1. Stop your on-premises PDC domain.
2. In your **override-values.yaml** file for **oc-cn-op-job-helm-chart**, set the following keys to match the PDC and cross-reference database schema details for your existing on-premises release:

```
ocpdc:
    configEnv:
        crossRefSchemaPDCTableSpace:
        crossRefSchemaTempTableSpace:
        crossRefSchemaUserName: UserName
        pdcSchemaPDCTableSpace:
        pdcSchemaTempTableSpace:
        pdcSchemaUserName: UserName
    secretValue:
        crossRefSchemaPassword: EncryptedPassword
        pdcSchemaPassword: EncryptedPassword
```

3. Set any other mandatory keys in your **override-values.yaml** file. For more information, refer to "Adding PDC Keys for oc-cn-op-job-helm-chart".
4. Deploy the PDC 16.26.0 domain to your cloud native environment:

```
helm install OpJobReleaseName oc-cn-op-job-helm-chart --namespace
BrmNameSpace --values OverrideValuesFile
```

   where:

   - *OpJobReleaseName* is the release name for **oc-cn-op-job-helm-chart** and is used to track this installation instance. It must be different from the one used for the BRM Helm chart.
   - *BrmNameSpace* is the namespace in which to create BRM Kubernetes objects for the BRM and Job Helm charts.
   - *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **oc-cn-op-job-helm-chart/values.yaml** file.
5. In your **override-values.yaml** file for **oc-cn-helm-chart**, set the **ocpdc.configEnv.upgrade** key to **true**.
6. Deploy the PDC 16.26.0 cloud native services to your cloud native environment, and ensure that the **pdc-domain** pod is in completed status:

```
helm install BrmReleaseName oc-cn-helm-chart --namespace BrmNameSpace --
values OverrideValuesFile
```

   where:

   - *BrmReleaseName* is the release name for **oc-cn-helm-chart** and is used to track this installation instance. It must be different from the one used for **oc-cn-op-job-helm-chart**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- *OverrideValuesFile* is the path to a YAML file that overrides the default configurations in the **oc-cn-helm-chart/values.yaml** file.

7. Ensure all files are upgraded successfully. To do so, check the status of the SQL upgrade log files in pdc-brm-pvc. If there is no failure message, all files upgraded successfully.

Copyright © 2019, 2026, Oracle and/or its affiliates.

21

Rolling Back Your BRM Database Upgrade

Learn how to roll back your Oracle Communications Billing and Revenue Management (BRM) database to previous release.

Topics in this document:

- About Rolling Back Your BRM Database
- Tasks for the Database Rollback Process
- About Reusing Your Previous BRM Installation

About Rolling Back Your BRM Database

The rollback feature allows you to revert your BRM database and other supported components to their previous stable state if issues occur after an upgrade. For example, if you upgrade your database from BRM 12.0.0.8.0 to BRM 16.26.0 and encounter problems, you can roll it back to BRM 12.0.0.8.0.

> **Note**
>
> You can only roll back your BRM database to the exact version that existed before the upgrade. For instance, if you upgrade from BRM 15.0 to 16.26.0, you can only roll back to 15.0. You cannot roll back to 15.1 or 12.0.0.*x*.0.

Before rolling back your BRM database, carefully review all prerequisites. Avoid making changes to system keys between the upgrade and rollback, and ensure you complete any rollbacks before making further schema or environment changes. In particular:

- Do not add secondary schemas to your BRM database before performing a rollback. Adding a schema first may hinder the rollback process.
- Do not rotate root keys between the upgrade and rollback.
- Do not add new optional managers to your existing BRM setup between the upgrade and rollback.

The BRM database rollback feature has the following limitations:

- Rolling back is supported only for BRM 16.26.0 or later. You cannot roll back from BRM 15.1 or earlier versions.
- Rolling back is limited to BRM 12.0.0.4.0 or later. You cannot revert to earlier releases.
- Rolling back can be performed only immediately after an upgrade.
- Rolling back is not supported for full BRM installations. To revert after a full installation, you must reinstall the earlier version of the software.
- After rolling back from BRM 16.26.0 to BRM 12.*x*, attempting to read BRM 16.26.0 events in BRM 12.*x* fails if the event version supported in BRM 12.*x* is lower than the version used in BRM 16.26.0. Because no automatic event downgrade occurs, you must manually

Copyright © 2019, 2026, Oracle and/or its affiliates.

convert the BRM 16.26.0 event format to the BRM 12.*x* event version before these events can be accessed in the rolled-back environment.

After rolling back your BRM database, you can:

- Roll back the BRM application so you can start using the previous BRM release. See "About Reusing Your Previous BRM Installation".
- In case you want to upgrade your BRM database again, you can upgrade to the same release you had rolled back from or a higher version.

  For example, if you roll back from BRM 16.26.0 to BRM 15.0, you can subsequently upgrade again to BRM 16.26.0 or later, but you cannot upgrade to BRM 15.1.

Tasks for the Database Rollback Process

The following lists the tasks required to roll back your BRM database to a previously installed release version.

> **Note**
>
> These instructions cover the *BRM database* rollback procedure. To roll back the BRM applications, run a Helm upgrade from the previous installation's directories.

1. In your **override-values.yaml** file for your 16.26.0 version of **oc-cn-init-db-helm-chart**, set the following keys:
   - **ocbrm.is_rollback**: Set this to **true**.
   - **ocbrm.is_upgrade**: Set to this to **false**.
2. Run the **helm install** command for the 16.26.0 version of **oc-cn-init-db-helm-chart**:

   ```
   helm install NewInitDbReleaseName oc-cn-init-db-helm-chart --values
   OverrideValuesFile --namespace BrmNameSpace
   ```

   where:
   - *NewInitDbReleaseName* is a new release name.
   - *OverrideValuesFile* is the file name and path to your **override-values.yaml** file.
   - *BrmNameSpace* is the namespace for **oc-cn-helm-chart**.
3. Verify that the database rollback completed successfully by checking that the ROLLBACK_STATUS field in the BRM_PS_T table is set to **1**.

About Reusing Your Previous BRM Installation

Before you can start using your previous BRM installation, perform these steps:

1. Roll back your BRM applications. To do so, run the **helm upgrade** command for the release you rolled back to. For example, if you rolled back from BRM 16.26.0 to BRM 15.0, run the **helm upgrade** command for the BRM 15.0 release.

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values PreviousOverrideValuesFile --
   namespace PreviousBrmNameSpace
   ```

   where:

Copyright © 2019, 2026, Oracle and/or its affiliates.

- *BrmReleaseName* is the release name used for the previous **oc-cn-helm-chart** install.
- *PreviousOverrideValuesFile* is the file name and path to your **override-values.yaml** file for the previous release.
- *PreviousBrmNameSpace* is the namespace that you used for your previous release.

2. Run the **load_pin_notify** utility through a configuration job.

> **Note**
>
> All steps must use the release you rolled back to. For example, if you rolled back from BRM 16.26.0 to BRM 15.0, use the BRM 15.0 versions of the files, directories, Helm chart, and scripts.

a. Add the following lines to the **oc-cn-helm-chart/config_scripts/loadme.sh** script:

```
#!/bin/sh

cd /oms/sys/data/config; load_pin_notify -v /oms/sys/data/config/pin_notify_file
exit 0;
```

where *pin_notify_file* is the name of the notification file.

b. Move the *pin_notify_file* input file to the **oc-cn-helm-chart/config_scripts** directory.

c. In the **override-values.yaml** file for **oc-cn-helm-chart**, set **ocbrm.config_jobs.run_apps** to **true**.

d. Run the **helm upgrade** command for the release your rolled back to.

```
helm upgrade BrmReleaseName oc-cn-helm-chart --values PreviousOverrideValuesFile
--namespace PreviousBrmNameSpace
```

e. Restart the CM by setting these keys in the **override-values.yaml** file and then run the **helm upgrade** command again.

- **ocbrm.config_jobs.restart_count**: Increment the existing value by **1**
- **ocbrm.config_jobs.run_apps**: Set this to **false**

After successfully rolling back your BRM system, you can begin reusing your system. However, before launching BRM, creating accounts, or running billing operations, do the following:

Copyright © 2019, 2026, Oracle and/or its affiliates.

Part VI

Troubleshooting BRM Cloud Native Deployments

This part provides information about troubleshooting issues that may occur while deploying Oracle Communications Billing and Revenue Management (BRM) cloud native in your system. It contains the following chapters:

- Troubleshooting Your BRM Cloud Native Deployment

Copyright © 2019, 2026, Oracle and/or its affiliates.

22

Troubleshooting Your BRM Cloud Native Deployment

Learn how to solve problems that may occur after the installation or upgrade of your Oracle Communications Billing and Revenue Management (BRM) cloud native system.

Topics in this document:

- Problems with the Helm Installation
- Helm Installation Fails with Time-Out Error
- BRM Cloud Native Deployment Out of Memory Errors
- PDC Messages Stuck in Rating Engine Queues
- PDC Interceptor Pod is Started But Went to Error State
- eceTopology.conf Errors While Restarting Pods
- SyncPDC Pod Fails to Start and Enters CrashLoopBackOff State
- Billing Care Web Services Token Authentication Fails After Custom KeyStore Configuration
- CM Fails During Deployment with Bad Login Error for /service/pcm_client

Problems with the Helm Installation

If a Helm installation encounters errors, such as an incorrect namespace, follow these steps to get back to a state where you can fix the issue and do a new installation.

> **Note**
>
> For more information about Kubernetes commands, see "kubectl Cheat Sheet" in the Kubernetes documentation.

1. Check the state of the deployment:

   ```
   kubectl get pods --output wide --namespace NameSpace
   ```

   To see information about a specific pod:

   ```
   kubectl describe pod PodName --namespace NameSpace
   ```

2. Use the **helm rollback** command to go back to a previous revision of the chart, or use the **helm uninstall** command to uninstall the chart. See "Rolling Back A Release To A Previous Revision" in *BRM Cloud Native System Administrator's Guide*, or see "Helm Uninstall" in the Helm documentation.
3. If neither rolling back nor uninstalling the chart are successful, do the following to identify Kubernetes resources that did not install correctly and then delete them:
   - Check and delete all other stateful set components from the cluster:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
kubectl get sts
```

If you identify a stateful set that you want to delete, scale the number of replicas:

```
kubectl scale statefulsets StatefulSetName --replicas=n
```

where *StatefulSetName* is the name of a stateful set, and *n* is the number of replicas you are scaling to. For more information, see "Scale a StatefulSet" in the Kubernetes documentation.

Then, delete the stateful set:

```
kubectl delete StatefulSetName
```

You can run **kubectl get sts** again to verify the deletions.

- If you need to clean up Apache Kafka and Apache ZooKeeper, scale to 0 and then delete:

```
kubectl scale sts/kafka_pod --replicas=0
kubectl scale sts/zookeeper_pod --replicas=0
kubectl get pods
kubectl get sts
kubectl delete sts kafka_pod zookeeper_pod
```

- If necessary, check any PVC, Secret, ConfigMap, or service that was created by the deployment. If the output from any of these commands shows something that you want to clean up, you can use **kubectl delete** to remove it.

  For example:

```
kubectl get pvc --all-namespaces
kubectl delete pvc PVCName

kubectl get secrets --all-namespaces
kubectl delete secret SecretName

kubectl get configmap --all-namespaces
kubectl delete configmap ConfigMapName

kubectl get svc --all-namespaces
kubectl delete svc SVC1 SVC2
```

Helm Installation Fails with Time-Out Error

After you deploy a Helm chart, you may receive the following error message indicating that the Helm chart installation failed:

```
Error: failed post-install: timed out waiting for the condition
```

This occurs because a post-installation job took longer than five minutes to complete.

To resolve the issue:

1. Purge your Helm release:

   ```
   helm delete BrmReleaseName --purge
   ```

   This removes and purges all resources associated with the last revision of the release.
2. Run the **Helm install** command again.

Copyright © 2019, 2026, Oracle and/or its affiliates.

If that does not fix the problem, increase the amount of time Kubernetes waits for a command to complete by including the **--timeout** *duration* argument with the **helm install** command. For example, to set the timeout duration to 10 minutes, you would enter this command:

```
helm install BrmReleaseName oc-cn-helm-chart --namespace BrmNameSpace --timeout 10m --
values OverrideValuesFile
```

BRM Cloud Native Deployment Out of Memory Errors

After you deploy BRM cloud native, you may receive an error message similar to the following:

```
ERROR: cm_cache_heap_malloc: name="fm_bparams_cache" - out of memory, size
requested=2216,high val=960
cm_cache_flist: PIN_ERR_NO_MEM:requested=2216, used=121456, allocated=122880, chunk=30,
cache name="fm_bparams_cache"
```

To resolve the issue:

1. In your **oc-cn-helm-chart** directory, open your CM ConfigMap file **configmap_pin_conf_cm.yaml**.
2. Add the following **fm_bparams_cache** entry to the file:

   ```
   - cm_cache fm_bparams_cache 40,245760,23
   ```

3. Save and close the file.
4. Run the **helm** upgrade command for **oc-cn-helm-chart**:

   ```
   helm upgrade BrmReleaseName oc-cn-helm-chart --values OverrideValuesFile --namespace
   BrmNameSpace
   ```

PDC Messages Stuck in Rating Engine Queues

Occasionally, PDC messages and changesets may become stuck in the rating engine queues.

To resolve the issue, delete both the RRE and BRE pods by running the following command:

```
kubectl --namespace BrmNameSpace delete pod PdcPodName
```

where:

- *BrmNameSpace* is the namespace in which the BRM Kubernetes objects reside.
- *PdcPodName* is the name of the pod.

Kubernetes automatically restarts the deleted pod, which restarts the transformation engine. Messages should start flowing again.

PDC Interceptor Pod is Started But Went to Error State

After you deploy PDC, the Interceptor pod may start but immediately transition to an error state.

This may occur because the RCU prefix is configured incorrectly. To find out if this is the case, run the following command:

```
kubectl describe domain DomainName --namespace NameSpace
```

If the issue is related to the RCU prefix, you will see something similar to the following:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
WLSDPLY-12409: createDomain failed to create the domain: Failed to get FMW
infrastructure database defaults from the service table : Got exception when auto
configuring the schema component(s) with data obtained from shadow table:
Failed to build JDBC Connection object:
```

To resolve the issue:

1. Make sure that the RCU prefix is configured successfully as part of **oc-cn-op-job-helm-chart**. To do so, run the following command:

   ```
   kubectl get pod --namespace BrmNameSpace
   ```

   If it is configured correctly, **pdc-configure-rcu-***xxxxx* will show a Completed status.

2. Make sure that the RCU prefix and password configured in the **override-values.yaml** file for **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart** matches, and that a valid host name, port, and service name have been configured in the **values.yaml** file.

If the values are not configured properly, do the following:

1. Uninstall PDC and then set the **ocpdc.isEnabled** key to **false** in your **override-values.yaml** file for **oc-cn-helm-chart**.
2. Run the **Helm upgrade** command for **oc-cn-helm-chart**.

   Wait until the PDC pods have stopped.
3. In the **override-values.yaml** file for **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**, configure the **ocpdc.configEnv.rcuPrefix** key and set the **ocpdc.isEnabled** key to **true**.
4. Run the **helm upgrade** command for **oc-cn-helm-chart** and **oc-cn-op-job-helm-chart**.

   Wait until the PDC pods are in Running status.

For more information about troubleshooting pod errors, see "Troubleshooting" in *Oracle WebLogic Kubernetes Operator Samples*.

eceTopology.conf Errors While Restarting Pods

While restarting the pricingupdater and brmgateway pods in your ECE cloud native deployment, you may receive an error message similar to the following:

```
ERROR MonitorFrameworkMessagesBundle-31300: Failed to initialize grid manager based on
topology file: eceTopology.conf property file:
ece.propertiesjava.lang.IllegalArgumentException: Nodes on a given host must be assigned
unique JMX ports
(check nodes: '[PricingUpdater node pricingupdater-6d575bf75b-r5q2t on Host
pricingupdater.ece-server.cluster,
PricingUpdater node pricingupdater-fbb9d7fb7-kqmxv on Host pricingupdater.ece-
server.cluster]')
```

This occurs because ECE cloud native has written invalid entries to the **eceTopology.conf** property file during the startup process.

To resolve the issue, do not restart the pricingupdater and brmgateway pods. Instead, scale down and then scale up those pods.

For example, to scale down and scale up the brmgateway pod:

1. Scale down the brmgateway pod to **0**:

   ```
   kubectl --namespace BrmNameSpace scale deploy brmgateway1 --replicas=0
   ```

   Wait for the brmgateway pod to stop.

Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Scale back up the brmgateway pod to **1**:

   ```
   kubectl --namespace BrmNameSpace scale deploy brmgateway1 --replicas=1
   ```

SyncPDC Pod Fails to Start and Enters CrashLoopBackOff State

After a fresh deployment of PDC 15.x, the **SyncPDC** pod deploys and the **SyncPDC** job runs, but the pod fails and enters a **CrashLoopBackOff** state.

This may occur because the deployment fails to create the **WorkEntity** table. The **pdc_domain** logs show that the table is missing.

If this issue occurs, the **SyncPDC** log file shows an error similar to the following:

```
oracle.communications.brm.pdc.syncpdc.SyncPDCJob execute
SEVERE: Exception executing SyncPDC job
javax.ejb.EJBException: Problem finding error class; nested exception is:
java.lang.ClassNotFoundException: Failed to load class
javax.persistence.PersistenceException
```

If you try to create the table manually and see the following error, you can see the reason that the deployment failed to create the table:

```
ORA-60019: Creating initial extent of size 14 in tablespace of extent size 5
```

To resolve the issue, update the PDC schema as follows:

1. Delete the existing PDC tablespace.
2. Create a new PDC tablespace with higher values for the **STORAGE** parameters.

   The following is a sample SQL*Plus command for creating a tablespace for PDC:

   ```
   create tablespace pdctables datafile '/PINDB/pdctables.dbf'
   size 600M reuse autoextend on default
   STORAGE (initial 1M next 1M pctincrease 0);
   ```

   > **Note**
   >
   > Choose the datafile name, location, size and storage values (/PINDB/pdcfiles.dbf, 600M and 1M in the example above) based on your environment.

   For more information about creating tablespaces, see "Creating Tablespace Storage" in *BRM Installation Guide*.
3. Redeploy PDC cloud native.

Billing Care Web Services Token Authentication Fails After Custom KeyStore Configuration

After upgrading Billing Care REST API pods with custom self-signed Identity and Trust KeyStores, Billing Care REST API requests authenticated using OAuth tokens may fail with the following error:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
Invalid Token/Missing Token
```

This issue may occur because the Billing Care REST API pods are using an incorrect or incompatible Oracle Identity Cloud Service (IDCS) public certificate in the Trust KeyStores, causing token validation to fail even when a valid access token is provided.

To resolve the issue:

1. Verify that the correct certificates are configured in both Identity and Trust KeyStores.
2. Ensure the certificate is obtained directly from the IDCS instance:
   a. Retrieve the IDCS certificate by entering this in your browser:

      https://*identityURL***/.well-known/idcs-configuration**

      where *identityURL* represents the Identity Service URL.
   b. Export the certificate from the browser.
      i. Open site information (look icon in address bar).
      ii. View certificate.
      iii. Export or download the certificate.
   c. Import the IDCS public certificate into the TrustStore:

      keytool -importcert -trustcacerts -keystore *keystoreName* **-file** *idcsCertificate*

      where:

      - *keystoreName* is the name of the custom TrustStore file that contains the certificates.
      - *idcsCertificate* is the public certificate exported from the IDCS instance.
3. Enable request and payload logging on the service to verify the Authorization header and token propagation.
4. Redeploy or upgrade the relevant jobs and application charts.

> **Note**
>
> This resolution is applicable only if there is an issue with IDCS public certificate.

CM Fails During Deployment with Bad Login Error for /service/pcm_client

When deploying Oracle Communications Billing and Revenue Management (BRM) 12 Patch Set 8 Cloud Native Environment (CNE) against an on-premises database upgraded from BRM 7.5 Patch Set 18 by using the BRM 12 Patch Set 8 upgrade scripts, the dm-oracle pod can start successfully, but Connection Manager (CM) can fail during startup with a bad login error for **/service/pcm_client**, such as PIN_ERR_BAD_LOGIN_RESULT.

This issue can occur when the BRM cloud native deployment does not use the same BRM root-key wallet and encryption key as the upgraded database schema. For example, if the upgraded schema contains an encrypted **/service/pcm_client** password, but the dm-oracle pod is started with a different BRM_CRYPT_KEY, CM authentication for root.0.0.0.1 can fail.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Ensure that the cloud native deployment uses the same BRM root-key wallet and encryption key as the source schema.

To fix this, do the following:

1. Get the encryption key value from the CRYPT_KEY_T table in the upgraded database.
2. Update the **brm_crypt_key** entry in the **oms-schema-password** Kubernetes Secret with the value from CRYPT_KEY_T.
3. Run a Helm upgrade. The Helm upgrade restarts all affected pods.
4. Run testnap and use login_pw to verify that CM can authenticate successfully with the updated encryption key or password.

Copyright © 2019, 2026, Oracle and/or its affiliates.