Oracle® Communications

Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide

Release 24.3.0
G13557-01
October 2024

Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide, Release 24.3.0

G13557-01

Contents

1 Introduction

1.1 Overview 1
1.2 References 2
1.3 Oracle Error Correction Policy 3
1.4 Oracle Open Source Support Policies 3

2 Installing UDR

2.1 Prerequisites 1
2.1.1 Software Requirements 1
2.1.2 Environment Setup Requirements 2
2.1.2.1 Client Machine Requirements 2
2.1.2.2 Network Access Requirement 2
2.1.2.3 Server or Space Requirement 3
2.1.2.4 Access to OpenStack Environment 3
2.1.2.5 Availability of a Public Key 3
2.1.2.6 UDR Software 3
2.1.2.7 CNE Requirement 3
2.1.2.8 cnDBTier Requirement 4
2.1.2.9 OSO Requirement 4
2.1.2.10 CNC Console Requirements 4
2.1.3 Resource Requirements 4
2.1.3.1 Default Resource Requirements for UDR 5
2.1.3.2 Minimum Resource Requirements for UDR 7
2.1.3.3 Performance 25K Diameter 9
2.1.3.4 Default Resource Requirements for SLF 10
2.1.3.5 Minimum Resource Requirements for SLF 11
2.1.3.6 Default Resource Requirements for EIR 13
2.1.3.7 Minimum Resource Requirements for EIR 15
2.1.3.8 Resource Requirements for UDR Hooks 16
2.1.3.9 Resource Requirements for UDR Tools 19
2.2 Installation Sequence 19
2.2.1 PreInstallation Tasks 19
2.2.1.1 Downloading Unified Data Repository Package 19

2.2.1.2 Pushing the Images to Customer Docker Registry 20
2.2.1.3 Verifying and Creating Namespace 27
2.2.1.4 Creating Service Account, Role, and RoleBinding 28
2.2.1.5 Pod Security Policy, Role, and Role Binding Creation 30
2.2.1.6 Configuring cnDBTier 33
2.2.1.7 Configuring Multiple Site Deployment 35
2.2.1.8 Creating Kubernetes Secret - DBName, Username, Password, and Encryption Key 36
2.2.1.9 Enabling HTTP1.1 38
2.2.1.10 Enabling HTTPS on Ingress Gateway and Egress Gateway 39
2.2.1.11 Creating Kubernetes Secret - Keys and Certificates for OAuth2 Support 42
2.2.1.12 Creating Persistent Volume Claim 43
2.2.1.13 UDR Compatibility with Kubernetes, CNE and Kyverno Policies 44
2.2.1.14 Configuring UDR to Support Aspen Service Mesh 44
2.2.1.15 Configuring udrServices and consumerNF Flags 68
2.2.1.16 Configuring eirServices and consumerNF Flags 71
2.2.1.17 Configuring PodDisruptionBudget Kubernetes Resource 72
2.2.1.18 Configuring Network Policies 74
2.2.1.19 Manual Creation of UDR Database and MySQL User 82
2.2.1.20 Secondary Path for PCF Notifications When the Primary PCF is Unavailable 84
2.2.1.21 Configuring HPA and PDB in UDR Custom Value File 87
2.2.1.22 Enabling and Disabling Conflict Resolution 88
2.2.2 Installation Tasks 88
2.2.2.1 Installing UDR Package 88
2.2.3 Post Installation Task 89
2.2.3.1 Verifying Installation 89
2.2.3.2 Performing Helm Test 90
2.2.3.3 Taking a Backup 91

3 Customizing UDR

3.1 UDR Configuration Parameters 1
3.1.1 Global Configurable Parameters 1
3.1.2 nudr-drservice Microservice Parameters 30
3.1.3 nudr-dr-provservice Microservice Parameters 38
3.1.4 nudr-notify-service Microservice Parameters 45
3.1.5 nrf-client-service Microservice Parameters 55
3.1.6 nudr-config Microservice Parameters 71
3.1.7 nudr-config-server Microservice Parameters 78
3.1.8 nudr-diameterproxy Microservice Parameters 85
3.1.9 appinfo microservice parameters 93
3.1.10 nudr-perf-info Microservice Parameters 99

3.1.11 diam-gateway Parameters 104
3.1.12 ocudr-ingressgateway-sig Microservice Parameters 112
3.1.13 ocudr-ingressgateway-prov Microservice Parameters 139
3.1.14 ocudr-egressgateway Microservice Parameters 151
3.1.15 alternate-route Microservice Parameters 180
3.1.16 Database Resolution Auditor Service 192

4 Upgrading UDR

4.1 Supported Upgrade Paths 1
4.2 Preupgrade Tasks 1
4.3 Upgrade Task 1
4.4 Postupgrade Tasks 7

5 Rolling Back UDR

5.1 Supported Rollback Paths 1
5.2 Rollback Tasks 1
5.3 Postrollback Tasks 3

6 Uninstalling UDR

6.1 Uninstalling UDR Using Helm 1
6.2 Deleting Kubernetes Namespace 2
6.3 Removing UDR Database 2
6.4 Cleaning UDR Deployment 4

7 Fault Recovery

7.1 Overview 1
7.2 Impacted Areas 2
7.3 Prerequisites 3
7.4 Fault Recovery Scenarios 3
7.4.1 Scenario 1: Database Migration to a New Cluster 3
7.4.1.1 Scenario 1A: Configuration Database Migration 4
7.4.1.2 Scenario 1B: Configuration and Subscriber Database Migration 4
7.4.2 Scenario 2: Deployment Failure 5
7.4.3 Scenario 3: Database Corruption 5
7.4.3.1 Scenario 3A: Configuration Database Corruption 5
7.4.3.2 Scenario 3B: Subscriber Database Corruption 5
7.4.4 Scenario 4: Site Failure 7
7.4.4.1 Scenario 4A: Single or Multiple Site Failure 7

7.4.4.2 Scenario 4B: All Sites Failure 8
7.5 Manual Database Backup and Restore 9
7.5.1 Backing up the UDR Database 9
7.5.2 Restoring the UDR Database 10

A Creating Private Keys and Certificates for Ingress Gateway

B Rollback Instructions to Revert to 29.519 v15.3.0 for PCF Data

My Oracle Support

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

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

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

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

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

Acronyms

The following table provides information about the acronyms and the terminologies used in the document:

Table Acronyms and Terminologies

Field	Description
5G-AN	5G Access Network
5GC	5G Core Network
5G-GUTI	5G Globally Unique Temporary Identifier
5GS	5G System
AMF	Access and Mobility Management Function
API	Application Programming Interface
ASM	Aspen Service Mesh
AUSF	Authentication Server Function
CEA	Capability Exchange Answer
CER	Capability Exchange Request
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
CRD	Custom Resource Definition
CSV	Comma Separated Value
cnPCRF	Cloud Native Policy and Charging Rules Function
ECR	Mobile Equipment Identity Check Request
EIC	Equipment Identity Check
EGW	Egress Gateway
EIR	Equipment Identity Register
FQDN	Fully Qualified Domain Name
GPSI	Generic Public Subscription Identifier
GSM	Global System for Mobile communication
HPA	Horizontal Pod Autoscaler
HTTP	Hypertext Transfer Protocol
IE	Information Element
IGW	Ingress Gateway
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
JSON	JavaScript Object Notation
MME	Mobility Management Entity
MSISDN	Mobile Station Integrated Services Digital Network
NEF	Oracle Communications Cloud Native Core, Network Exposure Function
NF	Network Function
NRF	Oracle Communications Cloud Native Core, Network Repository Function
NSI ID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information

Table (Cont.) Acronyms and Terminologies

Field	Description
NSSF	Oracle Communications Cloud Native Core, Network Slice Selection Function
NSSP	Network Slice Selection Policy
PDB	PodDisruptionBudget
PDBA	Provisioning Database Application
PDBI	Provisioning Database Interface
PEI	Permanent Equipment Identifier
PCF	Policy Control Function
PCRF	Policy and Charging Rules Function
PSP	Pod Security Policy
PVC	Persistent Volume Claim
REST	Representational State Transfer
SBA	Service-Based Architecture (SBA)
SBI	Service Based Interface
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy
SH	Diameter Interface
SLF	Subscriber Location Function
SMF	Session Management Function
SOAP	Simple Object Access Protocol
SUPI	Subscription Permanent Identifier
TCP	Transmission Control Protocol
TPS	Transaction Per Second
UDM	Unified Data Management
UDR	Oracle Communications Cloud Native Core, Unified Data Repository
UE	User Equipment
VSA	Vendor Specific Attribute
UDSF	Unstructured Data Storage Function
XML	Extensible Markup Language

What's New in This Guide

This section introduces the documentation updates for Release 24.3.x.

Release 24.3.0 - G13557-01, October 2024

- **General Updates:**
  - Updated the release number to 24.3.0 in the entire document.
  - Updated the versions of multiple software and environment setup requirements in the Prerequisites section.
  - Added steps to configure the supported HPA and PDB versions in UDR. For more information, see Configuring HPA and PDB in UDR Custom Value File.
- **Installation Updates:**
  - Added the following parameters in the Global Configurable Parameters section:
    - `s13InterfaceEnable`
    - `eirConfigurations.defaultResponse`
    - `eirConfigurations.accessLogEnabled`
    - `eirConfigurations.defaultResponseIMEINotFound`
    - `eirConfigurations.imsiLookupFallbackEnabled`
  - Removed the following parameters in the nudr-drservice Microservice Parameters section:
    - `eirConfigurations.defaultResponse`
    - `eirConfigurations.accessLogEnabled`
    - `eirConfigurations.defaultResponseIMEINotFound`
  - Added `notification.retryNotificationExpiry` parameter in the nudr-notify-service Microservice Parameters section.
  - Removed `retryNotificationExpiry` parameter in the nudr-notify-service Microservice Parameters section.
  - Added `diameter.messageBuffer.queueSize` parameter in the nudr-diameterproxy Microservice Parameters section.
- **Upgrade, Rollback, and Uninstall Updates:**
  - Updated the upgrade paths in the Upgrading UDR section.
  - Updated the rollback paths in the Supported Rollback Paths section.

1 Introduction

This guide describes how to install or upgrade Oracle Communications Cloud Native Core, Unified Data Repository (UDR) in a cloud native environment.

> **Note**
>
> - This guide covers the installation instructions when Podman is the container platform with Helm as the Packaging Manager. For any other container platform, the operator must use the commands based on to their deployed container runtime environment.
> - `kubectl` commands can vary based on the platform deployment. Replace `kubectl` with Kubernetes environment-specific command line tool to configure Kubernetes resources through kube-api server. The instructions provided in this document are as per the CNE version of kube-api server.

> **Caution**
>
> User, computer and applications, and character encoding settings may cause an issue when copy-pasting commands or any content from PDF. PDF reader version also affects the copy-pasting functionality. It is recommended to verify the pasted content especially when the hyphens or any special characters are part of the copied content.

1.1 Overview

UDR is a key component of the 5G Service Based Architecture (SBA). It is implemented as a cloud native function and offers a unified database for storing application, subscription, authentication, service authorization, policy data, session binding, and application state information.

> **Note**
>
> The performance and capacity of the UDR system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment, including but not limited to, the size of the json payload, operation type, and traffic model.

It provides:

- an HTTP2 based RESTful interface and APIs to provision other Network Functions (NFs) data.
- provisioning clients to access stored data.

As per the 3GPP specifications, UDR supports the following functionality:

- Storage and retrieval of subscription data by UDM
- Storage and retrieval of policy data by PCF
- Storage and retrieval of structured data for exposure
- Storage and retrieval of SLF information consumed by NRF
- Application data (including Packet Flow Descriptions (PFDs) for application detection, AF request information for multiple UEs) by the NEF
- Subscription and Notification feature

UDR:

- leverages a common Oracle Communications Cloud Native Framework.
- is compliant with 3GPP 29.505 Release 15 specification UDM data access.
- is compliant with 3GPP 29.519 Rel 16(backward compatible with Rel 15) specification for PCF data access.
- is compliant with 3GPP 29.519 Rel16 specification for NEF data access.
- has tiered architecture providing separation between the connectivity, business logic and data layers.
- uses MySQL NDB Cluster CGE edition as the backend database in the Data Tier.
- registers with NRF in the 5G network, so the other NFs in the network can discover UDR through NRF.
- registers UDR with services: DR-SERVICE and GROUP-ID-MAP.

Unstructured Data Storage Function (UDSF):

- Supports storage and retrieval of unstructured data by any 5G NF. The specifications of UDSF is not defined by 3gpp.
- This functionality is part of Oracle's 5G UDR solution.

Subscriber Location Function (SLF):

- Supports Nudr-groupid-map service as defined by 3GPP
- Registers with NRF for Nudr-groupid-map service
- Is complaint with 3GPP Release 16 for APIs to be consumed by 5G NRF
- Supports REST or JSON based provisioning APIs for SLF data

5G Equipment Identity Register (EIR) supports:

- REST/JSON based provisioning of 5G EIR profile
- Bulk import of 5G EIR subscriber data via PDBI interface
- N5g-eir_EquipmentIdentityCheck Service API
- 5G EIR as per 3GPP TS 29.511

1.2 References

Refer to the following documents while deploying UDR:

- *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*
- *Oracle Communications Cloud Native Core, Provisioning Gateway Guide*

- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Network Repository Function, Upgrade, and Fault Recovery Guide.*
- *Oracle Communications Cloud Native Core, Network Function Data Collector User Guide*
- *Oracle Communications Cloud Native Core, cnDBTier User Guide*

1.3 Oracle Error Correction Policy

The table below outlines the key details for the current and past releases, their General Availability (GA) dates, the latest patch versions, and the end dates for the Error Correction Grace Period.

Table 1-1 Oracle Error Correction Policy

Release Number	General Availability (GA) Date	Latest Patch Version	Error Correction Grace Period End Date
3.24.3	October 2024	24.3.0	October 2025
3.24.2	October 2024	24.2.1	July 2025
3.24.1	April 2024	24.1.0	April 2025
3.23.4	December 2023	23.4.1	December 2024

> **Note**
>
> For a release, Sev1 and Critical Patch Update (CPU) patches are supported for 12 months. For more information, see Oracle Communications Cloud Native Core and Network Analytics Error Correction Policy.

1.4 Oracle Open Source Support Policies

Oracle Communications Cloud Native Core uses open source technology governed by the Oracle Open Source Support Policies. For more information, see Oracle Open Source Support Policies.

2
Installing UDR

This chapter provides information about installing Oracle Communications Cloud Native Core, Unified Data Repository (UDR) in a cloud native environment.

> **Note**
>
> UDR supports both fresh installation, and it can also be upgraded from 24.2.x. 24.1.x. and 23.4.x. For more information on how to upgrade UDR, see Upgrading Unified Data Repository.

2.1 Prerequisites

Before installing and configuring UDR, ensure that the following prerequisites are met:

2.1.1 Software Requirements

This section lists the software that must be installed before installing UDR.

Table 2-1 Preinstalled Software

Software	Versions
Kubernetes	1.30.0, 1.29.1, and 1.28.6
Helm	3.15.2, 3.13.2, and 3.12.3
Podman	4.2.0, 4.0.2, and 3.4.2

To check the versions of the preinstalled software in the cloud native environment, run the following commands:

```
kubectl version
```

```
helm version
```

```
Podman version
```

The following software are available if UDR is deployed in CNE. If you are deploying UDR in any other cloud native environment, these additional software must be installed before installing UDR. To check the installed software, run the following command:

```
helm3 ls -A
```

The list of additional software items, along with the supported versions and usage, is provided in the following table:

Table 2-2 Additional Software

Software	Version	Purpose
OpenSearch Dashboard	2.11.0	Logging
OpenSearch	2.11.0	Logging
containerd	1.7.16	Container Lifecycle Managemet
Fuentd Opensearch	1.16.2	Logging
MetalLB	0.14.4	Loadbalancer
Prometheus	2.52.0	Metrics and Alerts
Prometheus Operator	0.76.0	Metrics and Alerts
Grafana	9.5.3	Metrics and KPIs
Jaeger	1.60.0	Tracing
Istio	1.18.2	Service mesh
Kyverno	1.12.5	Logging
cert-manager	1.12.4	Certificates management
Velero	1.12.4	Migration, backup and restore for Kubernetes cluster

2.1.2 Environment Setup Requirements

This section describes the environment setup required for installing UDR.

2.1.2.1 Client Machine Requirements

This section describes the requirements for client machine, that is, the machine used by the user to run deployment commands.

The client machine should have:

- Helm repository configured.
- network access to the Helm repository and Docker image repository.
- network access to the Kubernetes cluster.
- required environment settings to run the `kubectl` and `docker`, and `Podman` commands. The environment should have privileges to create a namespace in the Kubernetes cluster.
- Helm client installed with the 'push' plugin. Configure the environment in such a manner that the `helm install` command to deploy the software in the Kubernetes cluster.

2.1.2.2 Network Access Requirement

The Kubernetes cluster hosts must have network access to the following repositories:

- Local Helm repository: It contains the UDR Helm charts. To check if the Kubernetes cluster hosts can access the local Helm repository, run the following command:

  ```
  helm repo update
  ```

- Local docker image repository: It contains the UDR docker images. To check if the Kubernetes cluster hosts can access the local Docker image repository, pull any image with an image-tag using either of the following commands:

  ```
  podman pull <Podman-repo>/<image-name>:<image-tag>
  ```

  Where:

  – `<Podman-repo>` is the IP address or host name of the Podman repository.
  – `<image-name>` is the Docker image name.
  – `<image-tag>` is tag assigned to the Docker image used for the UDR pod.

> **Note**
>
> Run the `kubectl` and `helm` commands on a system based on the deployment infrastructure. For instance, it can be run on a client machine such as VM, server, local desktop, and so on.

2.1.2.3 Server or Space Requirement

For information about server or space requirements, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

2.1.2.4 Access to OpenStack Environment

The user should have access to an existing OpenStack environment, including the OpenStack desktop. This environment is configured with appropriate resource flavors and network resources that allow its users to allocate resources to the virtual machines.

2.1.2.5 Availability of a Public Key

The user must have a public key for logging into the Bootstrap Host. This key should be placed into the customer OpenStack environment using the **Import Key** tab on the **Launch Instance**→**Key Pair** dialog box or through the **Compute**→**Access and Security**.

2.1.2.6 UDR Software

For UDR software, the user must install Kubernetes v1.30.0 and HELM v3.14.2. The UDR software consists of:

- Helm Charts that reflect the UDR software version. It is a zipped tar file.
- Docker images of the microservices that are shared as tar file.

Tools Package

It consists of deployment template yaml files for the nudr-migration and nudr-bulk-import services.

2.1.2.7 CNE Requirement

This section is applicable only if you are installing UDR on Cloud Native Environment (CNE).

UDR supports CNE 24.3.x, 24.2.x, and 24.1.x.

To check the CNE version, run the following command:

```
echo $OCCNE_VERSION
```

For more information about CNE, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

2.1.2.8 cnDBTier Requirement

UDR supports cnDBTier 24.3.x, 24.2.x, and 24.1.x. cnDBTier must be configured and running before installing UDR. For more information about cnDBTier installation, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

For more information about the cnDBTier customizations required for UDR and SLF, see the following files:

- `ocudr_udr_35msub_N3617.2K_SH10K_TPS_dbtier_custom_values_24.3.0-rc.5.yaml`
- `ocudr_udr_35msub_N36_20K_TPS_dbtier_custom_values_24.3.0-rc.5.yaml`
- `ocudr_slf_50msub_Lookup_34K_Prov_14.4K_TPS_dbtier_custom_values_24.3.0-rc.5.yaml`
- `ocudr_eir_1msub_10K_SH_10K_N17_TPS_dbtier_custom_values_24.3.0-rc.5.yaml`

2.1.2.9 OSO Requirement

UDR supports Operations Services Overlay (OSO) 24.3.x, 24.2.x, and 24.1.x, for common operation services (Prometheus and components such as alertmanager, pushgateway) on a Kubernetes cluster, which does not have these common services. For more information on installation procedure, see *Oracle Communications Cloud Native Core, Operations Services Overlay Installation and Upgrade Guide*.

2.1.2.10 CNC Console Requirements

UDR supports CNC Console 24.3.x. For more information about CNC Console, see *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide.*.

2.1.3 Resource Requirements

This section lists the resource requirements to install and run UDR, SLF, EIR, UDR Hooks, and UDR Tools. For more information about performance and resource requirements, see *Oracle Communications Cloud Native Core, Unified Data Repository Benchmarking Guide.*

> **Note**
>
> The performance and capacity of the UDR system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment, including but not limited to, the size of the json payload, operation type, and traffic model.

Table 2-3 Resource Profile for UDR Deployment

Resource Requirement	Minimum	Minimum (Including side car requirement)	Default	Default (Including side car requirement)	Performance-2 5K Diameter Only
CPU	26 CPUs	48 CPUs	64 CPUs	102 CPUs	71 CPUs
Memory	26 GB	48 GB	64CPUs	102 GB	86 GB
Ephemeral Storage	21 GB	21 GB	38 GB	38 GB	23 GB
Individual Microservice Resource Requirements	Minimum Resource Requirements for UDR	Minimum Resource Requirements for UDR	Default Resource Requirements for UDR	Default Resource Requirements for UDR	Performance 25K Diameter

For information about resources required for UDR hooks and UDR tools, see:

- Resource Requirements for UDR Hooks
- Resource Requirements for UDR Tools

Table 2-4 Resource Profile for SLF Deployment

Resource Requirement (Includes side car requirement)	Minimum	Minimum (Including side car requirement)	Default	Default (Including side car requirement)
CPU	22 CPUs	40 CPUs	47 CPUs	79 CPUs
Memory	18 GB	36 GB	47 GB	78 GB
Ephemeral Storage	18 GB	18 GB	30 GB	30 GB
Individual Microservice Resource Requirements	Minimum Resource Requirements for SLF	Minimum Resource Requirements for SLF	Default Resource Requirements for SLF	Default Resource Requirements for SLF

Table 2-5 Resource Profile for EIR Deployment

Resource Requirement (Includes side car requirement)	Minimum	Minimum (Including side car requirement)	Default	Default (Including side car requirement)
CPU	22 CPUs	40 CPUs	47 CPUs	80 CPUs
Memory	18 GB	36 GB	47 GB	79 GB
Ephemeral Storage	18 GB	18 GB	30 GB	30 GB
Individual Microservice Resource Requirements	Minimum Resource Requirements for EIR	Minimum Resource Requirements for EIR	Default Resource Requirements for EIR	Default Resource Requirements for EIR

2.1.3.1 Default Resource Requirements for UDR

The following table shows the default resource requirements to deploy UDR:

Table 2-6 Default Resource Requirements: UDR with Aspen Service Mesh (ASM) Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU	Total Memory	Total Ephemeral Storage
nudr-drservice	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-dr-provservice	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-config	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-config-server	2 CPUs	2 GB	1 GB	2	2 CPUs	4 GB	2 GB
ingressgateway-sig	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
ingressgateway-prov	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
egressgateway	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-diameterproxy	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
diam-gateway	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-ondemandmigration	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-notify-service	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
alternate-route	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
perf-info	1 CPU	1 GB	1 GB	2	2 CPU	2 GB	2 GB
app-info	0.5 CPU	1 GB	1 GB	2	2 CPU	2 GB	2 GB
nudr-nrf-client-nfmanagement	1 CPU	1 GB	1 GB	2	2 CPU	2 GB	2 GB
nudr-dbcr-auditor-service	2 CPUs	2 GB	1 GB	1	2 CPU	2 GB	1 GB
Total	-	-	-	-	**CPU**	**Memory**	**Ephemeral Storage**
Total resources (Considering Main service containers)	-	-	-	-	56 CPUs	56 GB	31 GB

Table 2-6 (Cont.) Default Resource Requirements: UDR with Aspen Service Mesh (ASM) Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU	Total Memory	Total Ephemeral Storage
Additional Resources (Hooks/Init/ Update Containers)	-	-	-	-	8 CPUs	8 GB	7 GB
Total Resources	-	-	-	-	64 CPUs	64 GB	38 GB
Total side car resources (Applicable for ASM setups)	-	-	-	-	38 CPUs	38 GB	NA
Grand Total for UDR with side car requirement	-	-	-	-	102 CPUs	102 GB	38 GB

> **Note**
>
> If the debug tool is enabled, then the additional resource requirements are 1 CPU, 2GB RAM, and 4G ephemeral storage for each pod.

2.1.3.2 Minimum Resource Requirements for UDR

The minimum resource requirements to install UDR with ASM enabled are:

Table 2-7 Minimum Resource Requirements: UDR with Aspen Service Mesh (ASM) Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per minimum deployment)	Total CPU	Total Memory	Total Ephemeral Storage
nudr-drservice	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
nudr-dr-provservice	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
nudr-config	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB

Table 2-7 (Cont.) Minimum Resource Requirements: UDR with Aspen Service Mesh (ASM) Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per minimum deployment)	Total CPU	Total Memory	Total Ephemeral Storage
nudr-config-server	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
nudr-nrf-client-nfmanagement	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
ingressgateway-sig	2 CPUs	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
ingressgateway-prov	2 CPUs	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
egressgateway	2 CPUs	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
nudr-diameterproxy	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
diam-gateway	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
nudr-ondemand migration	1 CPU	1 GB	1 GB	1	1 CPUs	1 GB	1 GB
nudr-notify-service	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
alternate-route	2 CPU	1 GB	1 GB	1	2 CPU	2 GB	1 GB
perf-info	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
app-info	0.5 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
nudr-dbcr-auditor-service	1 CPUs	1 GB	1 GB	1	1 CPU	1 GB	1 GB
Total	-	-	-	-	**CPU**	**Memory**	**Ephemeral Storage**
Total UDR resources (Considering default values)	-	-	-	-	20 CPUs	20 GB	16 GB
Additional Resources (Hooks/Init/Update Containers)	-	-	-	-	6 CPUs	6 GB	5 GB
Total Resources	-	-	-	-	26 CPUs	26 GB	21 GB

Table 2-7 (Cont.) Minimum Resource Requirements: UDR with Aspen Service Mesh (ASM) Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per minimum deployment)	Total CPU	Total Memory	Total Ephemeral Storage
Total side car resources (Applicable for ASM setups)	-	-	-	-	22 CPUs	22 GB	NA
Grand Total for SLF with side car	-	-	-	-	48 CPUs	48 GB	21 GB

> **Note**
>
> If the debug tool is enabled, then the additional resource requirements are 1 CPU, 2GB RAM, and 4G ephemeral storage for each pod.

2.1.3.3 Performance 25K Diameter

The resources required to deploy diameter based microservices are:

Table 2-8 Resources for Diameter Based Microservices

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage per pod	Replicas	Total CPU	Total Memory	Total Ephemeral Storage
nudr-diam-gateway	6 CPUs	4 GB	1 GB	2	12 CPUs	8 GB	2 GB
nudr-diameterproxy	3 CPUs	4 GB	1 GB	19	57 CPUs	76 GB	19 GB
Additional Resources (Hooks/Init/Update Containers)	-	-	-	-	2 CPUs	2 GB	2 GB
Total Resources	-	-	-	-	71 CPUs	86 GB	23 GB

2.1.3.4 Default Resource Requirements for SLF

Table 2-9 Default Resource Requirements: SLF with ASM enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Sidecar container CPU Per Pod	Sidecar container Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU (Micro service + Sidecar Containers)	Total Memory for the Pod (Micro service + Sidecar Containers)	Total Ephemeral Storage
nudr-drservice	2 CPUs	2 GB	1 CPUs	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
nudr-dr-provservice	2 CPUs	2 GB	1 CPUs	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
nudr-config	2 CPUs	2 GB	1 CPU	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
nudr-config-server	2 CPUs	2 GB	1 CPU	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
nudr-nrf-client-nfmanagement	1 CPU	1 GB	1 CPU	1 GB	1 GB	2	4 CPUs	4 GB	2 GB
ingressgateway-sig	2 CPUs	2 GB	1 CPUs	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
ingressgateway-prov	2 CPUs	2 GB	1 CPUs	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
egressgateway	2 CPUs	2 GB	1 CPUs	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
alternate-route	2 CPUs	2 GB	1 CPU	1 GB	1 GB	2	6 CPUs	6 GB	2 GB
perf-info	1 CPU	1 GB	1 CPU	1 GB	1 GB	2	4 CPUs	4 GB	2 GB
app-info	0.5 CPU	1 GB	1 CPU	1 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-dbcr-auditor-service	1 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
Total	-	-	-	-	-	-	**CPU**	**Memory**	**Ephemeral Storage**
SLF resource Requirement	-	-	-	-	-	-	39 CPUs	39 GB	23 GB

Table 2-9 (Cont.) Default Resource Requirements: SLF with ASM enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Sidecar container CPU Per Pod	Sidecar container Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU (Micro service + Sidecar Containers)	Total Memory for the Pod (Micro service + Sidecar Containers)	Total Ephemeral Storage
Additional Resources (Hooks/ Init/ Update Containers)	-	-	-	-	-	-	8 CPUs	8 GB	7 GB
SLF Total Resources	-	-	-	-	-	-	47 CPUs	47 GB	30 GB
Total side car resources	-	-	-	-	-	-	32 CPUs	31 GB	NA
Total Resources	-	-	-	-	-	-	79 CPUs	78 GB	30 GB

Note

If the debug tool is enabled, then the additional resource requirements are 1 CPU, 2GB RAM, and 4G ephemeral storage for each pod.

2.1.3.5 Minimum Resource Requirements for SLF

The resource requirements to install SLF with ASM enabled are:

Table 2-10 Minimum Resource Requirements: SLF with ASM Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Sidecar container CPU Per Pod	Sidecar container Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU (Microservice + Sidecar Containers)	Total Memory for the Pod (Microservice + Sidecar Containers)	Total Ephemeral Storage
nudr-drservice	1 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
nudr-dr-provservice	1 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
nudr-config	1 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
nudr-config-server	1 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
nudr-nrf-client-nfmanagement	1 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
ingressgateway-sig	2 CPUs	1 GB	1 CPU	1 GB	1 GB	1	3 CPUs	2 GB	1 GB
ingressgateway-prov	2 CPUs	1 GB	1 CPU	1 GB	1 GB	1	3 CPUs	2 GB	1 GB
egressgateway	2 CPUs	1 GB	1 CPU	1 GB	1 GB	1	3 CPUs	2 GB	1 GB
alternate-route	2 CPU	1 GB	1 CPU	1 GB	1 GB	1	3 CPUs	2 GB	1 GB
perf-info	1 CPU	1GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
app-info	0.5 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
nudr-dbcr-auditor-service	1 CPU	1 GB	1 CPU	1 GB	1 GB	1	2 CPUs	2 GB	1 GB
Total	-	-	-	-	-	-	**CPU**	**Memory**	**Ephemeral Storage**
SLF resource Requirement	-	-	-	-	-	-	16 CPUs	12 GB	12 GB

Table 2-10 (Cont.) Minimum Resource Requirements: SLF with ASM Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Sidecar container CPU Per Pod	Sidecar container Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU (Micro service + Sidecar Containers)	Total Memory for the Pod (Micro service + Sidecar Containers)	Total Ephemeral Storage
Additional Resources (Hooks/ Init/ Update Containers)	-	-	-	-	-	-	6 CPUs	6 GB	6 GB
SLF Total Resources	-	-	-	-	-	-	22 CPUs	18 GB	18 GB
Total side car resources	-	-	-	-	-	-	18 CPUs	18 GB	NA
Total Resources	-	-	-	-	-	-	40 CPUs	36 GB	18 GB

> **Note**
>
> If the debug tool is enabled, then the additional resource requirements are 1 CPU, 2GB RAM, and 4G ephemeral storage for each pod.

2.1.3.6 Default Resource Requirements for EIR

Table 2-11 Default Resource Requirements: EIR

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU	Total Memory for the Pod	Total Ephemeral Storage
nudr-drservice	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-dr-provservice	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-config	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB

Table 2-11 (Cont.) Default Resource Requirements: EIR

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU	Total Memory for the Pod	Total Ephemeral Storage
nudr-config-server	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
nudr-nrf-client-nfmanagement	1 CPU	1 GB	1 GB	2	4 CPUs	4 GB	2 GB
ingressgateway-sig	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
ingressgateway-prov	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
egressgateway	2 CPUs	2 GB	1 GB	2	4 CPUs	4 GB	2 GB
alternate-route	2 CPUs	2 GB	1 GB	2	4 CPUs	2 GB	2 GB
perf-info	1 CPU	1 GB	1 GB	2	2 CPUs	2 GB	2 GB
app-info	0.5 CPU	1 GB	1 GB	2	2 CPUs	2 GB	2 GB
nudr-dbcr-auditor-service	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
Total	-	-	-	-	**CPU**	**Memory**	**Ephemeral Storage**
EIR resource Requirement	-	-	-	-	39 CPUs	39 GB	23 GB
Additional Resources (Hooks/Init/Update Containers)	-	-	-	-	8 CPUs	8 GB	7 GB
EIR Total Resources	-	-	-	-	47 CPUs	47 GB	30 GB
Total side car resources, if installed with ASM	-	-	-	-	33 CPUs	32 GB	NA
Total Resources	-	-	-	-	80 CPUs	79 GB	30 GB

Note

If the debug tool is enabled, then the additional resource requirements are 1 CPU, 2GB RAM, and 4G ephemeral storage for each pod.

2.1.3.7 Minimum Resource Requirements for EIR

Table 2-12 Minimum Resource Requirements: EIR

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per minimum deployment)	Total CPU	Total Memory for the Pod	Total Ephemeral Storage
nudr-drservice	1 CPU	1 GB	1 GB	1	1 CPUs	1 GB	1 GB
nudr-dr-provservice	1 CPU	1 GB	1 GB	1	1 CPUs	1 GB	1 GB
nudr-config	1 CPU	1 GB	1 GB	1	1 CPUs	1 GB	1 GB
nudr-config-server	1 CPU	1 GB	1 GB	1	1 CPUs	1 GB	1 GB
nudr-nrf-client-nfmanagement	1 CPU	1 GB	1 GB	1	1 CPUs	1 GB	1 GB
ingressgateway-sig	2 CPUs	1 GB	1 GB	1	2 CPUs	1 GB	1 GB
ingressgateway-prov	2 CPUs	1 GB	1 GB	1	2 CPUs	1 GB	1 GB
egressgateway	2 CPUs	1 GB	1 GB	1	2 CPUs	1 GB	1 GB
alternate-route	2 CPU	1 GB	1 GB	1	2 CPUs	1 GB	1 GB
perf-info	1 CPU	1GB	1 GB	1	1 CPUs	1 GB	1 GB
app-info	0.5 CPU	1 GB	1 GB	1	1 CPUs	1 GB	1 GB
nudr-dbcr-auditor-service	1 CPU	1 GB	1 GB	1	1 CPU	1 GB	1 GB
Total	-	-	-	-	**CPU**	**Memory**	**Ephemeral Storage**
EIR resource Requirement	-	-	-	-	16 CPUs	12 GB	12 GB
Additional Resources (Hooks/Init/Update Containers)	-	-	-	-	6 CPUs	6 GB	6 GB
EIR Total Resources	-	-	-	-	22 CPUs	18 GB	18 GB
Total side car resources, if installed with ASM	-	-	-	-	18 CPUs	18 GB	NA

Table 2-12 (Cont.) Minimum Resource Requirements: EIR

Microservice Name	CPU Per Pod	Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per minimum deployment)	Total CPU	Total Memory for the Pod	Total Ephemeral Storage
Total Resources	-	-	-	-	40 CPUs	36 GB	18 GB

> **Note**
>
> If the debug tool is enabled, then the additional resource requirements are 1 CPU, 2GB RAM, and 4G ephemeral storage for each pod.

2.1.3.8 Resource Requirements for UDR Hooks

For UDR hooks, the following set of resources should be available:

Table 2-13 Install, Upgrade, Rollback, Delete or Uninstall, and Test Hooks

Hook Name	CPU Requirement	Memory Requirement	Side car CPU Requirement	Side car Memory Requirement	Ephemeral Storage Requirement
nudr-config-server-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-config-server-post-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-config-server-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-config-server-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-config-server-pre-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-config-server-post-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-config-server-pre-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-config-server-post-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-post-install	1 CPU	1 GB	1 CPU	1 GB	1 GB

Table 2-13 (Cont.) Install, Upgrade, Rollback, Delete or Uninstall, and Test Hooks

Hook Name	CPU Requirement	Memory Requirement	Side car CPU Requirement	Side car Memory Requirement	Ephemeral Storage Requirement
nudr-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
ingressgateway-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
ingressgateway-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
ingressgateway-pre-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
ingressgateway-pre-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
ingressgateway-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
ingressgateway-post-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
ingressgateway-pre-upgrade-oauth-validator	1 CPU	1 GB	1 CPU	1 GB	1 GB
egressgateway-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
egressgateway-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
egressgateway-pre-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
egressgateway-pre-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
egressgateway-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
egressgateway-post-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
alternate-route-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
alternate-route-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
alternate-route-pre-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
alternate-route-pre-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
alternate-route-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
alternate-route-post-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
perf-info-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
perf-info-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB

Table 2-13 (Cont.) Install, Upgrade, Rollback, Delete or Uninstall, and Test Hooks

Hook Name	CPU Requirement	Memory Requirement	Side car CPU Requirement	Side car Memory Requirement	Ephemeral Storage Requirement
perf-info-pre-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
perf-info-pre-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
perf-info-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
perf-info-post-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
app-info-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
app-info-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
app-info-pre-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
app-info-pre-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
app-info-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
app-info-post-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-nrf-client-nfmanagement-pre-install	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-nrf-client-nfmanagement-pre-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-nrf-client-nfmanagement-pre-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-nrf-client-nfmanagement-pre-delete	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-nrf-client-nfmanagement-post-upgrade	1 CPU	1 GB	1 CPU	1 GB	1 GB
nudr-nrf-client-nfmanagement-post-rollback	1 CPU	1 GB	1 CPU	1 GB	1 GB
test	1 CPU	1 GB	NA	NA	1 GB

Note

- These hooks are applicable for both SLF and UDR deployments with ASM. However, these hooks are executed on demand, for example, during install, uninstall, upgrade, rollback, or helm-test cases.
- All the hooks in the above table except test are cleared once the job is completed.

2.1.3.9 Resource Requirements for UDR Tools

The resource details to install UDR Tools are:

Table 2-14 UDR Tools with ASM Enabled

Microservice Name	CPU Per Pod	Memory Per Pod	Sidecar container CPU Per Pod	Sidecar container Memory Per Pod	Ephemeral Storage Per Pod	Replicas (As per default deployment)	Total CPU (Micro service + Sidecar Containers)	Total Memory for the Pod (Micro service + Sidecar Containers)	Total Ephemeral Storage
nudr-bulk-import	4 CPUs	6 GB	2 CPUs	1 GB	1 GB	1	6 CPUs	7 GB	1 GB
nudr-migration	4 CPUs	4 GB	2 CPUs	1 GB	1 GB	1	6 CPUs	5 GB	1 GB
nudr-export-tool	4 CPUs	3 GB	2 CPUs	1 GB	1 GB	1	6 CPUs	5 GB	1 GB

> **Note**
>
> If the Debug tool is enabled, then you need additional resources; 1 CPU, 2GB RAM and 4G ephemeral storage for each pod.

2.2 Installation Sequence

This section describes UDR preinstallation, installation, and postinstallation tasks.

2.2.1 PreInstallation Tasks

Before installing UDR, perform the tasks described in this section:

2.2.1.1 Downloading Unified Data Repository Package

To download the UDR package from My Oracle Support (MOS), perform the following steps:

1. Log in to My Oracle Support using your login credentials.
2. Select the **Patches and Updates** tab.
3. In the **Patch Search** console, select **Product or Family (Advanced)** option.
4. Enter `Oracle Communications Cloud Native Core - 5G` in **Product** field and select the product from the Product drop-down list.

5. From the **Release** drop-down list select **Oracle Communications Cloud Native Core Unified Data Repository <release_number>**. Where, <release_number> indicates the required release number of OCUDR.
6. Click **Search**. The **Patch Advanced Search Results** list appears.
7. Select the required patch from the search results. The **Patch Details** window appears.
8. Click **Download**. The **File Download** window appears.
9. Click the <p********_<release_number>_Tekelec>.zip file to download the package.

2.2.1.2 Pushing the Images to Customer Docker Registry

UDR deployment package includes ready-to-use images and Helm charts to orchestrate containers in Kubernetes.

To push the images to the registry:

1. Unzip the release package to the location where you want to install UDR. The package is as follows: `ReleaseName_pkg_Releasenumber.tgz`.
2. Untar the UDR package zip file to get UDR image tar file:
   `tar -xvf ocudr_pkg_24_3_0_0_0.tgz`

   This command results into `ocudr_pkg_24_3_0_0_0.tgz` directory. The directory consists of following:

   - ocudr_pkg_24_3_0_0_0.tgz
   - ocudr-24.3.0.tgz (helm chart)
   - ocudr-24.3.0.tgz.sha256 (checksum)
   - nudr-bulk-import-24.3.0.tgz (helm chart for bulk import tool)
   - nudr-bulk-import-24.3.0.tgz.sha256 (checksum)
   - nudr-export-tool-24.3.0.tgz (helm chart for export tool)
   - nudr-export-tool-24.3.0.tgz.sha256 (checksum)
   - nudr-migration-tool-24.3.0.tgz (helm chart for migration tool)
   - nudr-migration-tool-24.3.0.tgz.sha256 (checksum)
   - ocudr-servicemesh-config-24.3.0.tgz (service mesh helm chart)
   - nudr-export-tool-24.3.0.tgz.sha256 (checksum)
   - ocudr-servicemesh-config-24.3.0.tgz.sha256 (checksum)
   - ocudr-images-24.3.0.tar (docker images)
   - ocudr-images-24.3.0.tar.sha256 (checksum)
   - ocudr_custom_configtemplates_24.3.0.zip
   - ocudr_custom_configtemplates_24.3.0.zip.sha256
   - Readme.txt (Contains cksum and md5sum of tarballs)
3. Verify the package content from the `Readme.txt` file.

4. Run one of the following commands to load the ocudr-images-24.3.0.tar file into the Docker system. Run the following command:

```
docker load --input /root/ocudr-images-24.3.0.tar
```

```
sudo podman load --input /root/ocudr-images-24.3.0.tar
```

5. To verify if the image is loaded correctly, run the following command:
`docker images | grep ocudr`

> **Note**
>
> If required, re-tag and push the images according to their repository.

The following table lists the docker images for UDR:

Table 2-15 UDR Images

Service Name	Image	Image Version
nudr-drservice	ocudr/nudr_datarepository_service	24.3.0
	ocudr/debug-tools	24.3.1
nudr-dr-provservice	ocudr/nudr_datarepository_service	24.3.0
	ocudr/debug-tools	24.3.1
nudr-notify-service	ocudr/nudr_notify_service	24.3.0
	ocudr/debug-tools	24.3.1
nrf-client	ocudr/nrf-client	24.3.2
	ocudr/debug-tools	24.3.1
ingressgateway	ocudr/ocingress_gateway	24.3.3
	ocudr/configurationinit	24.3.3
	ocudr/configurationupdate	24.3.3
	ocudr/debug-tools	24.3.1
egressgateway	ocudr/ocegress_gateway	24.3.3
	ocudr/configurationinit	24.3.3
	ocudr/configurationupdate	24.3.3
	ocudr/debug-tools	24.3.1
nudr-config	ocudr/nudr_config	24.3.0
	ocudr/debug-tools	24.3.1
nudr-config-server	ocudr/ocpm_config_server	24.3.3
	ocudr/debug-tools	24.3.1
nudr-diameterproxy	ocudr/nudr_diameterproxy	24.3.0
	ocudr/debug-tools	24.3.1
diam-gateway	ocudr/nudr_diameterproxy	24.3.0
	ocudr/debug-tools	24.3.1
nudr-ondemand-migration	ocudr/nudr_ondemand_migration	24.3.0
	ocudr/debug-tools	24.3.1

Table 2-15 (Cont.) UDR Images

Service Name	Image	Image Version
alternate-route	ocudr/alternate_route	24.3.3
	ocudr/debug-tools	24.3.1
nudr-perf-info	ocudr/perf-info	24.3.3
	ocudr/debug-tools	24.3.1
nudr-app-info	ocudr/app-info	24.3.3
	ocudr/debug-tools	24.3.1
nudr-bulk-import	ocudr/nudr-bulk-import	24.3.0
	ocudr/nudr-xmltocsv	24.3.0
	ocudr/nudr-pdbitocsv	24.3.0
	ocudr/debug-tools	24.3.1
nudr-export-tool	ocudr/nudr-export-tool	24.3.0
	ocudr/debug-tools	24.3.1
nudr-migration	ocudr/nudr_pre_migration_hook	24.3.0
	ocudr/nudr_migration	24.3.0
	ocudr/debug-tools	24.3.1
nudr-dbcr-auditor-service	ocudr/nudr_dbcr_auditor_service	24.3.0
	ocudr/debug-tools	24.3.1
ingressgateway/egressgateway/alternate-route/perf-info/app-info hooks	ocudr/common_config_hook	24.3.3
nudr-pre-install-hook/nudr-pre-upgrade-hook/nudr-post-upgrade-hook/nudr-post-rollback-hook/nudr-post-install-hook	ocudr/nudr_common_hooks	24.3.0
nudr-bulk-import-pre-upgrade-hook	ocudr/nudr_bulk_import_pre_upgrade_hook	24.3.0
nudr-bulk-import-post-upgrade-hook	ocudr/nudr_bulk_import_post_upgrade_hook	24.3.0
nudr-export-tool-pre-upgrade-hook	ocudr/nudr_export_tool_pre_upgrade_hook	24.3.0
nudr-export-tool-post-upgrade-hook	ocudr/nudr_export_tool_post_upgrade_hook	24.3.0
test	ocudr/nf_test	24.3.2

> **Note**
>
> The nudr-notify-service and nudr-diameterproxy-service microservices are not required for SLF deployment and nudr-notify-service, nudr-diameterproxy, and nudr-ondemand-migration are not required for EIR deployment. So, disable these services by setting the 'enabled' value as 'false' under the nudr-diameterproxy and nudr-notity-service, and nudr-ondemand-migration sections of the custom values file. For more information, see UDR Configuration Parameters.

6. Run one of the following commands to tag the images to the registry:

```
docker tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
```

```
sudo podman tag <customer repo>/<imagen name>:<image version> <customer
repo>/<imagen name>:<image version>
```

Sample Tag Commands:

```
# podman tag ocudr/nudr_datarepository_service:24.3.0 <customer repo>/
nudr_datarepository_service:24.3.0
# podman tag ocudr/nrf-client:24.3.2 <customer repo>/nrf-client:24.3.2
# podman tag ocudr/nudr_common_hooks:24.3.0 <customer repo>/
nudr_common_hooks:24.3.0
# podman tag ocudr/nf_test:24.3.2 <customer repo>/nf_test:24.3.2
# podman tag ocudr/nudr_bulk_import:24.3.0 <customer repo>/
nudr_bulk_import:24.3.0
# podman tag ocudr/nudr_bulk_import_pre_install_hook:24.3.0 <customer
repo>/nudr_bulk_import_pre_install_hook:24.3.0
# podman tag ocudr/nudr_bulk_import_post_upgrade_hook:24.3.0 <customer
repo>/nudr_bulk_import_post_upgrade_hook:24.3.0
# podman tag ocudr/nudr_export_tool_pre_install_hook:24.3.0 <customer
repo>/nudr_export_tool_pre_install_hook:24.3.0
# podman tag ocudr/nudr_export_tool_post_upgrade_hook:24.3.0 <customer
repo>/nudr_export_tool_post_upgrade_hook:24.3.0
# podman tag ocudr/nudr_export_tool:24.3.0 <customer repo>/
nudr_export_tool:24.3.0
# podman tag ocudr/nudr_migration:24.3.0 <customer repo>/
nudr_migration:24.3.0
# podman tag ocudr/nudr_pre_migration_hook:24.3.0 <customer repo>/
nudr_pre_migration_hook:24.3.0
# podman tag ocudr/nudr_config:24.3.0 <customer repo>/nudr_config:24.3.0
# podman tag ocudr/ocpm_config_server:24.3.3 <customer repo>/
ocpm_config_server:24.3.3
# podman tag ocudr/ocingress_gateway:24.3.3 <customer repo>/
ocingress_gateway:24.3.3
# podman tag ocudr/ocegress_gateway:24.3.3 <customer repo>/
ocegress_gateway:24.3.3
# podman tag ocudr/configurationinit:24.3.3 <customer repo>/
configurationinit:24.3.3
# podman tag ocudr/configurationupdate:24.3.3 <customer repo>/
configurationupdate:24.3.3
# podman tag ocudr/debug-tools:24.3.1 <customer repo>/debug-tools:24.3.1
# podman tag ocudr/alternate_route:24.3.3 <customer repo>/
```

```
alternate_route:24.3.3
# podman tag ocudr/common_config_hook:24.3.3 <customer repo>/
commo_config_hook:24.3.3
# podman tag ocudr/perf-info:24.3.3 <customer repo>/perf-info:24.3.3
# podman tag ocudr/app-info:24.3.3 <customer repo>/app-info:24.3.3
# podman tag ocudr/nudr_dbcr_auditor_service:24.3.0 <customer repo>/
nudr_dbcr_auditor_service:24.3.0
```

Below microservices are applicable only in UDR mode.

```
# podman tag ocudr/nudr_notify_service:24.3.0 <customer repo>/
nudr_notify_service:24.3.0
# podman tag ocudr/nudr_diameterproxy:24.3.0 <customer repo>/
nudr_diameterproxy:24.3.0
# podman tag ocudr/nudr_diam_gateway:24.3.0 <customer repo>/
nudr_diam_gateway:24.3.0
# podman tag ocudr/nudr_ondemand_migration:24.3.0 <customer repo>/
nudr_ondemand_migration:24.3.0
```

Note

- <customer_repo> is the local docker registry on the cluster having Port Number, if registry has a port attached to it.
- For OCCNE, copy the package to bastion server and use localhost:5000 as CUSTOMER_REPO to tag the images and push to bastion docker registry.
- You may need to configure the Docker certificate before the push command to access customer registry via HTTPS, otherwise, docker push command may fail.

7. Run one of the following commands to push the images to the registry:

```
docker push <docker-repo>/<image-name>:<image-tag>
```

```
# sudo podman push <customer repo>/<imagen name>:<image version>
```

Note

Beginning from OC-CNE 1.9, Docker is replaced by Podman. Run the following sample Podman command to push the Docker image:

Sample Push Commands:

```
# podman push <customer repo>/readiness_check:24.3.0
# podman push <customer repo>/nudr_datarepository_service:24.3.0
# podman push <customer repo>/nrf-client:24.3.2
# podman push <customer repo>/nudr_common_hooks:24.3.0
# podman push <customer repo>/nf_test:24.3.2
# podman push <customer repo>/nudr_bulk_import:24.3.0
```

```
# podman push <customer repo>/nudr_bulk_import_pre_install_hook:24.3.0
# podman push <customer repo>/nudr_bulk_import_post_upgrade_hook:24.3.0
# podman push <customer repo>/nudr_export_tool_pre_install_hook:24.3.0
# podman push <customer repo>/nudr_export_tool_post_upgrade_hook:24.3.0
# podman push <customer repo>/nudr_export_tool:24.3.0
# podman push <customer repo>/nudr_migration:24.3.0
# podman push <customer repo>/nudr_pre_migration_hook:24.3.0
# podman push <customer repo>/nudr_config:24.3.0
# podman push <customer repo>/ocpm_config_server:24.3.3
# podman push <customer repo>/ocingress_gateway:24.3.3
# podman push <customer repo>/ocegress_gateway:24.3.3
# podman push <customer repo>/configurationinit:24.3.3
# podman push <customer repo>/configurationupdate:24.3.3
# podman push <customer repo>/debug-tools:24.3.1
# podman push <customer repo>/alternate_route:24.3.3
# podman push <customer repo>/common_config_hook:24.3.3
# podman push <customer repo>/perf-info:24.3.3
# podman push <customer repo>/app-info:24.3.3
# podman push <customer repo>/nudr_dbcr_auditor_service:24.3.0
```

Below microservices are applicable only in UDR mode.

```
# podman push <customer repo>/nudr_notify_service:24.3.0
# podman push <customer repo>/nudr_diameterproxy:24.3.0
# podman push <customer repo>/nudr_diam_gateway:24.3.0
# podman push <customer repo>/nudr_ondemand_migration:24.3.0
```

8. Run the following command to untar the Helm files:
   `tar -xvzf ocudr-24.3.0.tgz`
9. To download the ocudr_custom_configtemplates_24.3.0.zip, click the file available on My Oracle Support (MOS).
10. Unzip the template to get ocudr_custom_configtemplates_24.3.0.zip file that contains the following:
    - ocudr_custom_values_24.3.0.yaml - Custom values for UDR or SLF used during helm installation.
    - ocudr_nudr_bulk_import_custom_values_24.3.0.yaml - Custom value file for UDR bulk import tool.
    - ocudr_nudr_export_tool_custom_values_24.3.0.yaml - Custom value file for UDR export tool.
    - ocudr_nudr_migration_tool_custom_values_24.3.0.yaml - Custom value file for UDR migration tool.
    - ocudr_servicemesh_config_custom_values_24.3.0.yaml - Custom values for UDR service mesh charts used during helm installation.
    - commonConfig_rollback.py - Rollback scripts for Database Schema rollback.
    - RollBack_Schema_15.3.py - Rollback scripts for Schema rollback to 15.3 version specification.
    - ocslf_alerts_24.3.0.yaml - SLF Alerts template for CNE 1.8 and below. Applicable for OSO 1.6.x and 22.3.x versions.
    - ocslf_alerts_haprom_24.3.0.yaml - SLF Alerts CRD for CNE 1.9 and above. Applicable only for CNE deployments

- ocslf_dashboard_24.3.0.json - SLF KPI and metrics representation template that should be loaded on Grafana.
- ocslf_dashboard_haprom_24.3.0.json - SLF KPI and metrics representation template that should be loaded on Grafana (CNE 1.9 and Above).
- ocslf_kibanaconfig_24.3.0.json - SLF Kibana Template for logging that should be loaded on Kibana.
- ocudr_alerts_24.3.0.yaml - UDR Alerts CRD, Applicable only for CNE deployments.
- ocudr_dashboard_24.3.0.json - UDR KPI and metrics representation template that should be loaded on Grafana.
- ocslf_alerts_24.3.0.yaml - EIR Alerts CRD, Applicable only for CNE deployments.
- oceir_dashboard_24.3.0.json - EIR KPI and metrics representation template that should be loaded on Grafana.
- ocslf_mib_24.3.0.mib - SLF top level mib file, where the Objects and their datatypes are defined.
- ocslf_mib_tc_24.3.0.mib - SLF mib file, where the Objects and their data types are defined.
- ocudr_mib_24.3.0.mib - OCUDR top level mib file, where the Objects and their data types are defined.
- ocudr_mib_tc_24.3.0.mib - OCUDR mib file, where the Objects and their data types are defined.
- oceir_mib_24.3.0.mib - EIR top level mib file, where the Objects and their datatypes are defined.
- oceir_mib_tc_24.3.0.mib - EIR mib file, where the Objects and their data types are defined.
- toplevel.mib - Top level MIB file, where the Objects and their data types are defined.
- ocudr_nudr_config_api_24.3.0.yaml - Open API spec file.
- ocudr_mgm_api_24.3.0.yaml - Open API Spec for MGM APIs
- ocudr_udr_35msub_N3617.2K_SH10K_TPS_dbtier_custom_values_24.3.0-rc.5.yaml
- ocudr_udr_35msub_N36_20K_TPS_dbtier_custom_values_24.3.0-rc.5.yaml
- ocudr_slf_50msub_Lookup_34K_Prov_14.4K_TPS_dbtier_custom_values_24.3.0-rc.5.yaml
- ocudr_eir_1msub_10K_SH_10K_N17_TPS_dbtier_custom_values_24.3.0-rc.5.yaml
- /db_conflict_resolution_sqlfiles/EIR_mode_ndb_replication_insert_UPGRADE.sql
- /db_conflict_resolution_sqlfiles/SLF_mode_ndb_replication_insert_UPGRADE.sql
- /db_conflict_resolution_sqlfiles/ALL_mode_ndb_replication_insert.sql
- /db_conflict_resolution_sqlfiles/EIR_mode_ndb_replication_insert.sql
- /db_conflict_resolution_sqlfiles/SLF_mode_ndb_replication_insert.sql
- /db_conflict_resolution_sqlfiles/ALL_mode_ndb_replication_insert_UPGRADE.sql

> **Note**
>
> It is recommended to configure the Docker certificate before running the push command to access customer registry through HTTPS, otherwise docker push command may fail.

2.2.1.3 Verifying and Creating Namespace

This section explains how to verify and create a namespace in the system.

> **Note**
>
> This is a mandatory step, and must be performed before proceeding any further. The namespace created or verified in this step is used as an input for further steps in the installation procedure.

To verify and create a namespace:

1. Run the following command to verify if the required namespace already exists in the system:
   In the output of the above command, if the namespace exists, continue with the Naming Convention for Namespaces section.

   ```
   kubectl get namespace
   ```

2. If the required namespace is not available, create the namespace using the following command::

   ```
   kubectl create namespace <required namespace>
   ```

3. Update the global.nameSpace parameter in the ocudr-24.3.x-custom-values.yaml file with the namespace created in the previous step.
   For example, the following kubectl command create the namespace `ocudr kubectl create namespace ocudr`

   ```
   global: # NameSpace where secret is deployed nameSpace: ocudr
   ```

Naming Convention for Namespaces

The namespace should::

- start and end with an alphanumeric character.
- contain 63 characters or less.
- contain only alphanumeric characters or '-'.

> **Note**
>
> It is recommended to avoid using the prefix, `kube-` when creating a namespace. The prefix is reserved for Kubernetes system namespaces.

2.2.1.4 Creating Service Account, Role, and RoleBinding

This section is optional and it describes how to manually create a service account, role, and rolebinding. It is required only when customer needs to create a role, rolebinding, and service account manually before installing UDR.

> **Note**
>
> The secret(s) should exist in the same namespace where UDR is getting deployed. This helps to bind the Kubernetes role with the given service account.

Creating Service Account, Role, and RoleBinding

1. Run the following command to create a UDR resource file:

   ```
   vi <ocudr-resource-file>
   ```

   Example:

   ```
   vi ocudr-sample-resource-template.yaml
   ```

2. Update the `ocudr-sample-resource-template.yaml` with release specific information:

   > **Note**
   >
   > Make sure to update **<helm-release>** and **<namespace>** in the following template with respective UDR namespace and UDR helm release name.

   A sample template to update the ocudr-sample-resource-template.yaml file with is given below:

   ```
   #
   # Sample template start
   #
   apiVersion: v1
   kind: ServiceAccount
   metadata:
     name: <helm-release>-serviceaccount
     namespace: <namespace>
   ---

   apiVersion: rbac.authorization.k8s.io/v1
   kind: Role
   metadata:
     name: <helm-release>-role
     namespace: <namespace>
   rules:
   - apiGroups:
     - "" # "" indicates the core API group
     resources:
     - services
     - configmaps
   ```

```
  - pods
  - secrets
  - endpoints
  - persistentvolumeclaims
  verbs:
  - get
  - watch
  - list
  - update
- apiGroups:
  - apps
  resources:
  - deployments
  - statefulsets
  verbs:
  - get
  - watch
  - list
  - update
- apiGroups:
  - autoscaling
  resources:
  - horizontalpodautoscalers
  verbs:
  - get
  - watch
  - list
  - update
- apiGroups:
  - rbac.authorization.k8s.io
  resources:
  - roles
  - rolebindings
  verbs:
  - get
  - watch
  - list
  - update
- apiGroups:
  - monitoring.coreos.com
  resources:
  - prometheusrules
  verbs:
  - get
  - watch
  - list
  - update
 ---

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: <helm-release>-rolebinding
  namespace: <namespace>
roleRef:
  apiGroup: rbac.authorization.k8s.io
```

```
  kind: Role
  name: <helm-release>-role
subjects:
- kind: ServiceAccount
  name:  <helm-release>-serviceaccount
  namespace: <namespace>
#
# Sample template end
#
```

3. Run the following command to create the service account, role and role binding:

   ```
   kubectl -n <ocudr-namespace> create -f ocudr-sample-resource-template.yaml
   ```

   Example: `kubectl -n ocudr create -f ocudr-sample-resource-template.yaml`

4. Update the `serviceAccountName` parameter in the `ocudr-custom-values-24.3.x.yaml` file with the value updated in the name field under `kind: ServiceAccount`. For more information about the `serviceAccountName` parameter, see Global Configurable Parameters.

2.2.1.5 Pod Security Policy, Role, and Role Binding Creation

The Debug Tool provides third-party troubleshooting tools for debugging the runtime issues in a lab environment. The following tools are available to debug UDR issues:

- tcpdump
- ip
- netstat
- curl
- ping
- dig

Running Debug tool containers from CNE version 23.2.0

Follow the steps below to run the debug tool containers from CNE version 23.2.0:

- You must update Kyverno policy to allow running of debug tool containers from CNE version 23.2.0 onwards.
- Exclude the udr deployment namespace by editing the cluster policy `disallow-capabilities`. For more information, see UDR Compatibility with Kubernetes, CNE and Kyverno Policies.

Running Debug tool containers for previous CNE versions:

This section describes how to create a Pod Security Policy (PSP), role, and role binding resources. These resources are used to define the Role Based Access Control (RBAC) rules for debug container running as part of each pod.

This step is optional. It is required only when the debug tool is enabled and customer wants to create a role, role binding, and service account for it manually.

This role binding should be associated with the service account that you have created in Creating Service Account, Role, and RoleBinding section.

To create a PSP, role, and role binding resources:

1. Add the following sample template to the sample resource input yaml file (ocudr-debug-tool-rbac.yaml).

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: ocudr-debug-tool-psp
spec:
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - NET_ADMIN
  - NET_RAW
  fsGroup:
    ranges:
    - max: 65535
      min: 1
    rule: MustRunAs
  runAsUser:
    rule: MustRunAsNonRoot
  seLinux:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  volumes:
  - configMap
  - downwardAPI
  - emptyDir
  - persistentVolumeClaim
  - projected
  - secret

---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: <ocudr-helm-release-name>-debug-tool-role
  namespace: <ocudr-namespace>
rules:
- apiGroups:
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
  resourceNames:
  - ocudr-debug-tool-psp

---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: <ocudr-helm-release-name>-debug-tool-rolebinding
  namespace: <ocudr-namespace>
roleRef:
```

```
    apiGroup: rbac.authorization.k8s.io
    kind: Role
    name: <ocudr-helm-release-name>-debug-tool-role
subjects:
- kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: <helm-release>-serviceaccount
    namespace: <ocudr-namespace>
```

2. Run the following command to create the resources:

```
kubectl –n <ocudr-namespace> create -f ocudr-debug-tool-rbac.yaml
```

The debug container uses readOnlyRootFileSystem, allowPrivilegeEscalation, and allowedCapabilities parameters as part of PSP. Other parameters are mandatory for PSP creation and you can customize them as per the CNE environment. It is recommended to use the default values used in the same template.

The PSP parameters are as follows:

Table 2-16 PSP Parameters

Parameter	Description
apiVersion	It defines the versioned schema of this representation of an object.
kind	It is a string value representing the REST resource this object represents.
metadata	Standard object's metadata.
metadata.name	Name must be unique within a namespace.
spec	spec defines the policy enforced.
spec.allowPrivilegeEscalation	This parameter controls whether or not a user is allowed to set the security context of a container to true.
spec.allowedCapabilities	This parameter provides a list of capabilities that are allowed to be added to a container.
spec.fsGroup	It controls the supplemental group applied to some volumes. **RunAsAny** allows any fsGroup ID to be specified.
spec.runAsUser	It controls the user ID with which the containers are run. **RunAsAny** allows any runAsUser to be specified.
spec.seLinux	In this parameter, **RunAsAny** allows any seLinuxOptions to be specified.
spec.supplementalGroups	It controls the group IDs that containers can add. **RunAsAny** allows any supplementalGroups to be specified.
spec.volumes	This parameter provides a list of allowed volume types. The allowable values correspond to the volume sources that are defined when creating a volume.

The Role parameters are as follows:

Table 2-17 Role Parameters

Parameter	Description
apiVersion	It defines the versioned schema of this representation of an object.

Table 2-17 (Cont.) Role Parameters

Parameter	Description
kind	It is a string value representing the REST resource this object represents.
metadata	Standard object's metadata.
metadata.name	Name must be unique within a namespace.
metadata.namespace	Namespace defines the space within which each name must be unique.
rules	Rules holds all the PolicyRules for this Role.
apiGroups	APIGroups is the name of the APIGroup that contains the resources.
rules.resources	Resources is a list of resources this rule applies to.
rules.verbs	Verbs is a list of verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule.
rules.resourceNames	ResourceNames is an optional list of names to which the rule applies.

The role binding parameters are as follows:

Table 2-18 Role Binding Parameters

Parameter	Description
apiVersion	It defines the versioned schema of this representation of an object.
kind	It is a string value representing the REST resource this object represents.
metadata	Standard object's metadata.
metadata.name	Name must be unique within a namespace.
metadata.namespace	Namespace defines the space within which each name must be unique.
roleRef	RoleRef can reference a Role in the current namespace or a ClusterRole in the global namespace.
roleRef.apiGroup	APIGroup is the group for the resource being referenced
roleRef.kind	Kind is the type of resource being referenced
roleRef.name	Name is the name of resource being referenced
subjects	Subjects holds references to the objects, the role applies to.
subjects.kind	Kind of object being referenced. Values defined by this API group are "User", "Group", and "ServiceAccount".
subjects.apiGroup	APIGroup holds the API group of the referenced subject.
subjects.name	Name of the object being referenced.

2.2.1.6 Configuring cnDBTier

With cnDBTier, UDR facilitates automatic user creation with its pre-install hook. However, ensure that there is a privileged user on the NDB cluster, which has privileges similar to root user. This user should have necessary permissions to allow connections from remote hosts.

For cnDBTier version 24.3.x, enable the flag `ndb_allow_copying_alter_table` on cnDBTier configurations during installation.

Single Site Deployment

Perform the following steps on each of the SQL nodes.

1. Log in to MySQL on each of the API nodes of cnDBTier to verify this.

```
mysql>select host from mysql.user where User='<privileged username>';
+------+
| host |
+------+
| %    |
+------+
1 rowinset(0.00 sec)
```

2. If you do not see '%' in the output of the above query, modify this field to allow remote connections to root.

```
mysql>update mysql.user set host='%' where User='<privileged username>';
Query OK, 0rowsaffected (0.00 sec)
Rowsmatched: 1  Changed: 0  Warnings: 0
mysql> flushprivileges;
Query OK, 0rowsaffected (0.06 sec)
```

> **Note**
>
> Perform this step on each SQL node.

Multiple Site Deployment

To configure cnDBTier in case of multiple site deployment:

1. Update mysqld configuration in the cnDBTier custom values file before the installing or upgrading UDR or SLF.

```
global:
  ndbconfigurations:
    api
      auto_increment_increment: 3
      auto_increment_offset: 1
```

> **Note**
>
> - Set the `auto_increment_increment` parameter as same as number of sites. For example: If the number of sites is 2, set its value as 2 and if the number of sites is 3, set its value as 3.
> - Set the `auto_increment_offset` parameter as site ID. For example: The site ID for Site 1 is 1, Site 2 is 2, for Site 3 is 3 and so on.

2. If the fresh installation or upgrade of UDR or SLF on cnDBTier is not planned, then run the following command to edit the exiting mysqldconfig configmap on all the cnDBTier sites.

   ```
   kubectl edit configmap mysqldconfig <db-site-namespace>
   ```

   For Example: `kubectl edit configmap mysqldconfig -n site1`

   > **Note**
   >
   > Update the `auto_increment_increment` and `auto_increment_offset` values as mentioned in the previous step for all sites.

2.2.1.7 Configuring Multiple Site Deployment

In case of multiple site deployment of EIR or SLF, there is only one subscriber database which is used by each site and different configuration databases for each site, as each site has its own configuration. To have different configuration databases and same subscriber database, you need to create secrets accordingly. For more information about creating secrets, see Creating Kubernetes Secret - DBName, Username, Password, and Encryption Key.

To configure multiple site deployment:

1. Configure `nfInstanceId` under the `global` section of the ocudr-custom-values-24.3.0.yaml file differently for each UDR, SLF, or EIR site deployed.

   > **Note**
   >
   > Ensure the `nfInstanceId` configuration in the global section is same as that in the `appProfile` section of nrf-client.

```
global:
  # Unique ID to register to NRF, Should be configured differently on
multi site deployments for each SLF/UDR
  nfInstanceId: &nfInsId 5a7bd676-ceeb-44bb-95e0-f6a55a328b03

nrf-client:
  configmapApplicationConfig:
    profile: |-
      appProfiles=[{"nfInstanceId":"5a7bd676-ceeb-44bb-95e0-
f6a55a328b03","nfStatus":"REGISTERED","fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","nfType":"UDR","allowedNfTypes":
["NRF"],"plmnList":
[{"mnc":"14","mcc":"310"}],"priority":10,"capacity":500,"load":0,"locality"
:"bangalore","nfServices":[{"load":0,"scheme":"http","versions":
[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","ipEndPoints":
[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus
":"REGISTERED","allowedNfTypes":
["NRF"],"serviceInstanceId":"547d42af-628a-4d5d-
a8bd-38c4ba672682","serviceName":"nudr-group-id-
map","priority":10,"capacity":500}],"udrInfo":
{"groupId":"udr-1","externalGroupIdentifiersRanges":
```

```
[{"start":"10000000000","end":"20000000000"}],"supiRanges":
[{"start":"10000000000","end":"20000000000"}],"gpsiRanges":
[{"start":"10000000000","end":"20000000000"}]},"heartBeatTimer":90,"nfServi
cePersistence":false,"nfProfileChangesSupportInd":false,"nfSetIdList":
["setxyz.udrset.5gc.mnc012.mcc345"]}]
```

2. Configure `fullnameOverride` under the `config-server` section to <helm-release-name>-config-server. It should be different for each site deployed.

```
config-server:
  fullnameOverride: ocudr1-config-server
```

3. Configure **fullnameOverride** under the **appinfo** section to <helm-release-name>-app-info. It should be different for each site deployed.

```
appinfo:
  fullnameOverride: ocudr1-app-info
```

4. For cnDBTier configurations in multiple site deployment, see Configuring cnDBTier.

2.2.1.8 Creating Kubernetes Secret - DBName, Username, Password, and Encryption Key

UDR Database

For UDR application, two types of databases are required:

1. **Subscriber Database**: Subscriber database includes all subscriber data such as SLF, EIR, PCF, UDM, and NEF data.
2. **Configuration database**: Configuration database includes configuration content for all microservices.

UDR Users

This section explains how database administrators can create users and database in a single and multisite deployment.

> **Note**
>
> - Before running the procedure for georedundant sites, ensure that the cnDBTier for georedundant sites is already up and replication channels are enabled.
> - While performing a fresh installation, if UDR is already deployed, purge the deployment, and remove the database and users that were used for the previous deployment. For uninstallation procedure, see Uninstalling UDR.

There are two types of UDR database users with different set of permissions:

- **Privileged User**: This user requires to be granted remote access and complete set of permissions to create, alter, and drop udrdb and udrconfigdb databases across the SQL nodes. If necessary, the user should have the required permissions to create the application user.
- **Application User**: This user provided in the secret can be created by the UDR if global.preInstall.createUser=true in the custom-values.yaml. If the application user is

manually created, then the user requires to be granted remote access and have a minimum set of permission for the udrdb and udrconfigdb to insert, update, get, and remove the records across the SQL nodes.

Single Site Deployment

To create a secret, run the following command:

```
kubectl create secret generic ocudr-secrets --from-literal=dbname=<dbname> --
from-literal=configdbname=<configdbname> --from-
literal=privilegedUsername=<privilegedUsername> --from-
literal=privilegedPassword=<privilegedPassword> --from-
literal=dsusername=<udruserName> --from-literal=dspassword=<udruserPassword>
--from-literal=encryptionKey='My secret passphrase' -n <ocudr-namespace>
```

Example

```
kubectl create secret generic ocudr-secrets --from-literal=dbname=udrdb --
from-literal=configdbname=udrconfigdb --from-
literal=privilegedUsername=privUsr --from-
literal=privilegedPassword=privUsrPasswd --from-literal=dsusername=udruser --
from-literal=dspassword=udrpasswd --from-literal=encryptionKey='My secret
passphrase' -n myudr
```

Multiple Site Deployment

In case of multiple site deployment, configure the `configdbname` differently for each site. Each site has its own configuration content and it is stored separately in their respective configuration databases. Multiple UDRs under the same segment also have different configuration databases.

To create a secret in case of two site deployment, run the following commands:

1. For site 1:

   ```
   kubectl create secret generic ocudr-secrets --from-literal=dbname=<dbname>
   --from-literal=configdbname=<configdbname1> --from-
   literal=privilegedUsername=<privilegedUsername> --from-
   literal=privilegedPassword=<privilegedPassword> --from-
   literal=dsusername=<udruserName> --from-
   literal=dspassword=<udruserPassword> --from-literal=encryptionKey='My
   secret passphrase' -n <ocudr1-namespace>
   ```

   Example:

   ```
   kubectl create secret generic ocudr-secrets --from-literal=dbname=udrdb --
   from-literal=configdbname=udrconfigdb1 --from-
   literal=privilegedUsername=privUsr --from-
   literal=privilegedPassword=privUsrPasswd --from-literal=dsusername=udruser
   --from-literal=dspassword=udrpasswd --from-literal=encryptionKey='My
   secret passphrase' -n myudr
   ```

2. For site 2:

```
kubectl create secret generic ocudr-secrets --from-literal=dbname=<dbname>
--from-literal=configdbname=<configdbname2> --from-
literal=privilegedUsername=<privilegedUsername> --from-
literal=privilegedPassword=<privilegedPassword> --from-
literal=dsusername=<udruserName> --from-
literal=dspassword=<udruserPassword> --from-literal=encryptionKey='My
secret passphrase' -n <ocudr2-namespace>
```

Example:

```
kubectl create secret generic ocudr-secrets --from-literal=dbname=udrdb --
from-literal=configdbname=udrconfigdb2 --from-
literal=privilegedUsername=privUsr --from-
literal=privilegedPassword=privUsrPasswd --from-literal=dsusername=udruser
--from-literal=dspassword=udrpasswd --from-literal=encryptionKey='My
secret passphrase' -n myudr
```

2.2.1.9 Enabling HTTP1.1

By default, UDR, EIR or SLF deployment supports HTTP2 requests. To enable HTTP1, perform the following steps:

> **Note**
>
> - This configuration is required for CNC Console to perform provisioning on UDR or SLF deployment.
> - CNC Console communicates only to ingressgateway-prov service for provisioning. Ingressgateway-prov communicates over selected ingressgateway-prov.global.publicHttp1SignalingPort port (81 port by default) which is a HTTP1.1 specific port. This configuration is not required to be enabled for ingressgateway-sig.

- Enable the `enableIncomingHttp1` parameter under **ingressgateway** section in ocudr-custom-values file.
- Configure the port to be used for HTTP1 using `publicHttp1SignalingPort` under **global** section in ocudr-custom-values file.
- To enable HTTPS on HTTP1 port, enable the configuration `enableTLSIncomingHttp1` under `ingressgateway` section in ocudr-custom-values file.

> **Note**
>
> There are additional configurations to enable HTTPS on Ingress Gateway. For more information, see Enabling HTTPS on Ingress Gateway and Egress Gateway.

Sample template

```
ingressgateway-prov:
 global:
   .....
   .....
   # port on which UDR's API-Gateway service is exposed
   # If httpsEnabled is false, this Port would be HTTP/2.0 Port (unsecured)
   # If httpsEnabled is true, this Port would be HTTPS/2.0 Port (secured
SSL)
   publicHttpSignalingPort: 80
   publicHttpsSignalingPort: 443
   # HTTP1 Port
   publicHttp1SignalingPort: 81

 ......
 ......

 # Enable HTTP1
 enableIncomingHttp1: true

 # Enable Secure HTTP1
 enableTLSIncomingHttp1: true
```

- You are required to configure routes on cncc-core-custom-values file to connect to its port on ingressgateway for the provisioning routes configured. For this, check the `routesConfig` section as follows:
  CNCC Core routesConfig

```
ingress-gateway:
  routesConfig:
  ......
  ......
# uri should be configured with UDR/SLF ingressgateway fqdn with HTTP1
port. (Example: ocudr-ingressgateway.ocudr:81)
# 81 used above should be same as publicHttp1SignalingPort configured in
ocudr-custom-values.yaml file
  - id: udr_ingress
    uri: http://<FQDN>:<PORT>
    path: /nudr-dr-prov/**,/nudr-dr-mgm/**,/nudr-group-id-map-prov/**,/slf-
group-prov/**
```

2.2.1.10 Enabling HTTPS on Ingress Gateway and Egress Gateway

This step is optional. It is required only when SSL settings need to be enabled on Ingress Gateway and Egress Gateway microservices of UDR.

> **Note**
>
> For more information about the ssl section configuration, see UDR Configuration Parameters.

> **Important**
>
> If the certificates are not available, then create them following the instructions given in the Creating Private Keys and Certificates for Ingress Gateway section of this guide.

For information about TLS support, see "Support for TLS" section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

2.2.1.10.1 Enabling HTTPS on Ingress Gateway

To enable HTTPS on Ingress Gateway:

1. Enable `initssl` and `enableIncomingHttps` parameters in the `ingressgateway-sig` and `ingressgateway-prov`sections of the ocudr-custom-values.yaml file.
2. Configure the following details in the ssl section under ingressgateway
   - Configure the following details in the **ssl** section of `ingressgateway-sig` and `ingressgateway-prov` in the ocudr-custom-values.yaml file.
     - Kubernetes namespace
     - Kubernetes secret name holding the certificate details
     - Certificate information

```
# Configure this section to support TLS with ingress gateway
ssl:
  # TLS verison used
  tlsVersion: TLSv1.2,TLSv1.3

  # Secret Details for certificates
  privateKey:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    rsa:
      fileName: rsa_private_key_pkcs1.pem
    ecdsa:
      fileName: ecdsa_private_key_pkcs8.pem

  certificate:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    rsa:
      fileName: apigatewayrsa.cer
    ecdsa:
      fileName: apigatewayecdsa.cer

  caBundle:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    fileName: caroot.cer

  keyStorePassword:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    fileName: key.txt
```

```
  trustStorePassword:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    fileName: trust.txt

  initialAlgorithm: RS256
```

> **Note**
>
> If there are different sets of ssl certificates for the signaling and the provisioning interface, use two different secrets for `ingressgateway-sig` and `ingressgateway-prov`.

2.2.1.10.2 Enabling HTTPS on Egress Gateway

To enable HTTPS on Egress Gateway:

1. Enable the `outgoingHttps` parameter under `global`section of the ocudr-custom-values.yaml file.
2. Configure the following details in the `ssl` section of `egressgateway` in the ocudr-custom-values.yaml file.
   - Kubernetes namespace
   - Kubernetes secret name holding the certificate details
   - Certificate information

```
# Configure this section to support TLS with egress gateway
ssl:
  # TLS verison used
  tlsVersion: TLSv1.2,TLSv1.3
  initialAlgorithm: RS256

  # Secret Details for certificates
  privateKey:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    rsa:
      fileName: rsa_private_key_pkcs1.pem
    ecdsa:
      fileName: ecdsa_private_key_pkcs8.pem

  certificate:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    rsa:
      fileName: apigatewayrsa.cer
    ecdsa:
      fileName: apigatewayecdsa.cer

  caBundle:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
```

```
    fileName: caroot.cer

  keyStorePassword:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    fileName: key.txt

  trustStorePassword:
    k8SecretName: ocudr-gateway-secret
    k8NameSpace: ocudr
    fileName: trust.txt
```

2.2.1.11 Creating Kubernetes Secret - Keys and Certificates for OAuth2 Support

This section describes how to create Kubernetes secret to store keys and certificates for OAuth2 support. This step is optional. It is required only when you want to enable Oauth2 validation on UDR Ingress Gateway microservice.

Creating a Secret to Store Keys and Certificates for OAuth2

To create a secret for storing keys and certificates for OAuth2:

1. NRF creates access tokens using following private keys.
   ECDSA private key (Example: `ecdsa_private_key_pkcs8.pem`)

   RSA private key (Example: `rsa_private_key_pkcs1.pem`)
2. To validate these access tokens, create a secret and configure the certificates fetched from NRF into the ocudr-ingress-gateway. The certificates naming format should be:

   ```
   <nrfInstanceId>_<AlgorithmUsed>.crt (6faf1bbc-6e4a-4454-a507-
   a14ef8e1bc5c_ES256.crt
   ```
3. To create a secret:
   a. Log in to Bastion Host or a server that has `kubectl`.
   b. Run the following command to create a namespace for the secret:

      ```
      kubectl create namespace ocudr
      ```
   c. Run the following command to create kubernetes secret for NF access token validation:

      ```
      kubectl create secret generic oauthsecret --from-
      file=6faf1bbc-6e4a-4454-a507
      -a14ef8e1bc5c_ES256.crt -n ocudr
      ```

      > **Note**
      >
      > The file names in the above command are same as in Step 1.

   d. Run the following command to verify whether a secret is created successfully or not:

      ```
      kubectl describe secret oauthsecret -n ocudr
      ```

Updating an OAuth Token

To update an OAuth token:

1. Log in to Bastion Host or a server that has `kubectl`.
2. To update the secret with new or updated details:
   a. Run the following command to delete the secret:

      ```
      kubectl delete secret oauthsecret -n ocudr
      ```

   b. Update the certificates required from NRF.
   c. Run the following command to recreate the secret with updated details:

      ```
      kubectl create secret generic oauthsecret --from-file=0263663c-
      f5c2-4d1b-9170-f7b1a9116337_ES256.crt -n ocudr
      ```

2.2.1.12 Creating Persistent Volume Claim

> **Note**
>
> The bulk import Persistent Volume Claim (PVC) and xmltocsv container PVC are created automatically using helm.

To create a PVC:

1. Use the following sample template for previous release versions and save it as <file-name>.yaml file.
   PersistentVolumeClaim Bulk Import Service

   ```
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: bulkimportpersistentclaim
   spec:
     storageClassName: <Please Provide your StorageClass  Name>
     accessModes:
       - ReadWriteOnce
     resources:
       requests:
         storage: 1Gi
   ```

2. Run the following command to create a PVC.

   ```
   kubectl create -f <file-name>.yaml
   ```

3. Run the following command to verify whether a PVC is created or not.

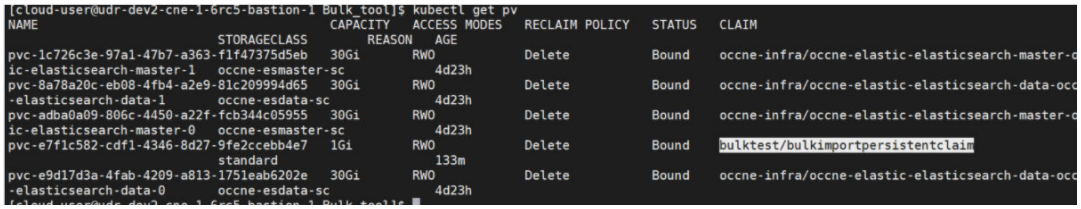
   ```
   kubectl get pvc -n <namespace>
   ```

Figure 2-1 Verifying PVC Creation

4. Run the following command to verify whether a Persistent Volume (PV) is created and bound to the PVC volume.

   ```
   kubectl get pv
   ```

Figure 2-2 Verifying PV Creation



For more information about Bulk Import Provisioning, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

2.2.1.13 UDR Compatibility with Kubernetes, CNE and Kyverno Policies

Perform the below steps before starting the ASM deployment to avoid ASM deployment failure due to Kubernetes version 1.30, CNE verion 24.3.x, and Kyverno policies.

```
kubectl patch clusterpolicy disallow-capabilities --type "json" -p
'[{"op":"add","path":"/spec/rules/0/exclude/any/0/resources/
namespaces/-","value":"testns"}] testns need to replace with user ns.
```

2.2.1.14 Configuring UDR to Support Aspen Service Mesh

UDR leverages the Platform Service Mesh (for example, Aspen Service Mesh (ASM)) for all internal and external TLS communication by deploying a special sidecar proxy in each pod to intercept all the network communications. The service mesh integration provides inter-NF communication and allows API gateway to co-work with service mesh. The service mesh integration supports the services by deploying a special sidecar proxy in each pods to intercept all the network communications between microservices.

The Service Mesh configurations are classified into:

- **Control Plane:** It involves adding labels or annotations to inject sidecar. The control plane configurations are already part of NF Helm chart.
- **Data Plane:** It helps in traffic management like handling NF call flows by adding Service Entries (SE), Destination Rules (DR), Envoy Filters (EF), and other resource changes like apiVersion change between versions. This is done manually depending on the NF requirement and Service Mesh deployment.

Data Plane Configuration

Data Plane configuration consists of following Custom Resource Definitions (CRDs):

- Service Entry (SE)
- Destination Rule (DR)
- Envoy Filter (EF)
- Peer Authentication
- Virtual Service (VS)
- Request Authentication (RA)
- Policy Authorization (PA)

> **Note**
>
> Use Helm charts to add or delete CRDs that you may require due to Service Mesh upgrades to configure features across different releases.

The Data Plane configuration is applicable in following scenarios:

- **NF to NF Communication:** During NF to NF communication where sidecar is injected on both NFs, you need SE and DR to communicate to the other NF, otherwise sidecar rejects the communication. All Egress communications of NFs must have an entry for SE and DR and the same needs to be configured.

  > **Note**
  >
  > For out of cluster communication, you must configure the core DNS with producer NF end point to enable the access.

- **Kube-api-server:** For Kube-api-server, there are few NF flows that require access to Kubernetes API server. The Service Mesh proxy (mTLS enabled) may block this. As per F5 recommendation, the NF need to add SE for Kubernetes API server for its own namespace.
- **Envoy Filters:** Sidecars rewrite the header with its own default value as a result the headers from back end services are lost. So, you need Envoy Filters to help in passing the headers from back end services to use it as it is.
- **Peer Authentication:** This template can be used to change the default mTLS mode on the deployment. It allows values such as STRICT, PERMISSIVE, and DISABLE.
- **Virtual Service:** If the target host returns 503 error response in the first attempt, then by default Istio tries to access the target host for two more times. To configure the number of retry attempts, use virtual service. Based on this configured number, Istio tries to access the target host.

  > **Note**
  >
  > If there is a requirement to configure multiple destination hosts, then it is recommended to use separate virtual services for each destination host instead of using one common virtual service.

- **Request Authentication**: This template is used to configure JWT tokens for OAuth use case. NF need to authenticate the OAuth token sent by consumer NFs. This is done by

using the public key of the NRF signing certificate and telling ServiceMesh to authenticate the token. UDR and SLF do not use this feature.

- **Policy Authorization**: Istio Authorization Policy enables access control on workloads in the mesh. Authorization policy supports CUSTOM, DENY and ALLOW actions for access control. When CUSTOM, DENY and ALLOW actions are used for a workload at the same time, the CUSTOM action is evaluated first, then the DENY action, and finally the ALLOW action is evaluated.

You need CRDs for the following UDR and SLF use cases:

- UDR to NRF communication (applicable for UDR and SLF)
- UDR to Notifications (applicable to UDR only)
- Kube-api-server (applicable to UDR and SLF)
- Envoy Filter for Serverheader and XFCC (applicable for UDR and SLF)

2.2.1.14.1 UDR Service Mesh Values.yaml File

A sample UDR Service Mesh values.yaml is given below:

> **Note**
>
> To connect to vDBTier, create an SE and DR for MySQL connectivity service if the database is in different cluster. Else, the sidecar rejects request as vDBTier does not support sidecars.

```
serviceEntries:
 # Required to communicate to k8 api server
  - hosts: |-
      [ "kubernetes.default.svc.cluster.local",
        "kubernetes.default.svc.cluster.local1" ]    # kubernetes cluster name
    exportTo: |-
      [ "." ]                     # refers to the current namespace
    location: MESH_INTERNAL
    addresses: |-
      [ "198.223.0.1",
        "198.223.0.2" ]
    ports:
    - number: 443
      name: https
      protocol: HTTPS
    name: kube-api-server

 # Add Service Entries for the producer NF's. Provide the public fqdn host
list along with the corresponding service entry related settings
  - hosts: |-
      [ "nrf123.oracle.com" ]         # pubic fqdn of destination NF (NRF)
    exportTo: |-
      [ "." ]
    location: MESH_EXTERNAL
    ports:
    - number: 8081
      name: http2-8081
      protocol: TCP
```

```
    - number: 8082
      name: http2-8082
      protocol: TCP
    resolution: NONE                    # should be NONE/DNS - if we are using
wildchars in fqdn it must be NONE
    name: nrf                          # unique name in service entry and
destination rule to identify destination NF

 # Below entry is required to send out notifications from UDR
  #- hosts: |-
  #    [ "notifyurl.oracle.com" ]
  #  location: MESH_EXTERNAL
  #  portno: 80
  #  portname: http2
  #  portprotocol: HTTP2
  #  resolution: NONE
  #  name: notify
  #  namespace: ocudr

destinationRules:
  - host: "nrf123.oracle.com"
    mode: DISABLE
    name: ocudr-to-other-nf-dr-test
    sbitimers: true
    tcpConnectTimeout: "750ms"
    tcpKeepAliveProbes: 3
    tcpKeepAliveTime: "1500ms"
    tcpKeepAliveInterval: "1s"
#  - host: "notifyurl.oracle.com"
#    mode: DISABLE
#    name: notify
#    sbitimers: false

# Add Envoy Filters for server header and XFCC filters w.r.t to asm version
1.6.x
#envoyFilters_v_16x:
#  - name: xfccfilter
#    labelselector: "app.kubernetes.io/instance: ocudr"
#    configpatch:
#      - applyTo: NETWORK_FILTER
#        filtername: envoy.filters.network.http_connection_manager
#        operation: MERGE
#        typeconfig: type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManager
#        configkey: forward_client_cert_details
#        configvalue: ALWAYS_FORWARD_ONLY
#  - name: serverheaderfilter
#    labelselector: "app.kubernetes.io/instance: ocudr"
#    configpatch:
#      - applyTo: NETWORK_FILTER
#        filtername: envoy.filters.network.http_connection_manager
#        operation: MERGE
#        typeconfig: type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManager
#        configkey: server_header_transformation
#        configvalue: PASS_THROUGH
```

```
# Add Envoy Filters for server header and XFCC filters w.r.t to asm version
1.9.x & 1.11.x
envoyFilters_v_19x_111x:
  - name: xfccfilter
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionMana
ger
        configkey: forward_client_cert_details
        configvalue: ALWAYS_FORWARD_ONLY
  - name: serverheaderfilter
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionMana
ger
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: custom-http-stream
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionMana
ger
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
        stream_idle_timeout: "6000ms"
        max_stream_duration: "7000ms"
        patchContext: SIDECAR_OUTBOUND
        networkFilter_listener_port: 8000
  - name: custom-tcpsocket-timeout
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: FILTER_CHAIN
        patchContext: SIDECAR_INBOUND
        operation: MERGE
        transport_socket_connect_timeout: "750ms"
        filterChain_listener_port: 8000
  - name: custom-http-route
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: HTTP_ROUTE
        patchContext: SIDECAR_OUTBOUND
        operation: MERGE
        route_idle_timeout: "6000ms"
        route_max_stream_duration: "7000ms"
```

```
        httpRoute_routeConfiguration_port: 8000
        vhostname: "ocudr.svc.cluster:8000"
  - name: logicaldnscluster
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: CLUSTER
        clusterservice: rchltxekvzwcamf-y-ec-
x-002.amf.5gc.mnc480.mcc311.3gppnetwork.org
        operation: MERGE
        logicaldns: LOGICAL_DNS
      - applyTo: CLUSTER
        clusterservice: rchltxekvzwcamd-y-ec-
x-002.amf.5gc.mnc480.mcc311.3gppnetwork.org
        operation: MERGE
        logicaldns: LOGICAL_DNS
  - name: slfexactbalancesample
    labelselector: "app.kubernetes.io/name: ingressgateway-prov"
    configpatch:
      - applyTo: LISTENER
        patchContext: SIDECAR_OUTBOUND
        listener_port: 5001
        operation: MERGE
        exactbalance: true

#PeerAuthentication is required to allow the incoming traffic with tls mode
peerAuthentication:
  - name: allservices
    labelselector: "app.kubernetes.io/part-of: ocudr"
    tlsmode: PERMISSIVE
#
# VirtualService is required to configure the retry attempts for the
destination host.
# By configuring this retries attempts will override the istio default
behaviour for retries for example for 503 response istio reties for 2 times.
# If we want mutiple virtualServices to configure, we can add entries like
below.
virtualService:
  - name: nrfvirtual1
    host: ocudr-ocudr-egress-gateway
    destinationhost: ocudr-ocudr-egress-gateway
    port: 8000
    exportTo: |-
      [ "." ]
    attempts: "0"
    timeout: 7s
  - name: nrfvirtual2
    host: ocudr-egress-gateway
    destinationhost: ocudr-egress-gateway
    port: 8000
    exportTo: |-
      [ "." ]
    retryon: 5xx
    attempts: "1"
    timeout: 7s

#
```

```
# RequestAuthentication is required to configure ouath secrets
# Note: Only one of jwks and jwksUri should be used as per istio document
#requestAuthentication:
#  - name: jwttokenwithjson
#    labelselector: "app: httpbin"
#    issuer: "jwtissue"
#    jwks: |-
#     '{
#       "keys": [{
#       "kid": "1",
#       "kty": "EC",
#       "crv": "P-256",
#       "x": "Qrl5t1-Apuj8uRI2o_BP9loqvaBnyM4OPTPAD_peDe4",
#       "y": "Y7vNMKGNAtlteMV-KJIaG-0UlCVRGFHtUVI8ZoXIzRY"
#      }]
#     }'
#  - name: jwttoken
#    labelselector: "app: httpbin"
#    issuer: "jwtissue"
#    jwksUri: https://example.com/.well-known/jwks.json

#authorizationPolicies:
#- name: allow-all-provisioning-on-ingressgateway-ap
#  labelselector: "app.kubernetes.io/name: ingressgateway"
#  action: "ALLOW"
#  hosts:
#    - "*"
#  paths:
#    - "/nudr-dr-prov/*"
#    - "/nudr-dr-mgm/*"
#    - "/nudr-group-id-map-prov/*"
#    - "/slf-group-prov/*"
#- name: allow-all-sbi-on-ingressgateway-ap
#  labelselector: "app.kubernetes.io/name: ingressgateway"
#  action: "ALLOW"
#  hosts:
#    - "*"
#  paths:
#    - "/nudr-group-id-map/v1/*"
#  xfccvalues:
#    - "*DNS=<dns-url-1>"
#    - "*DNS=<dns-url-2>"


# Template for Service Entry,Destination Rules and Envoy Filters
# The above range of host list will be used here in the below templates to
create CRD's
servicemeshResourcesTemplate: |
  {{- range .Values.serviceEntries }}
  apiVersion: networking.istio.io/v1alpha3
  kind: ServiceEntry
  metadata:
    name: {{ .name }}
    namespace: {{ .namespace | default $.Release.Namespace }}
  spec:
    hosts: {{ .hosts }}
```

```
  exportTo: {{ .exportTo }}
  {{ if .addresses }}
  addresses: {{ .addresses }}
  {{ end }}
  ports:
  {{- range .ports }}
  - number: {{ .number }}
    name: {{ .name }}
    protocol: {{ .protocol }}
  {{- end }}
  location: {{ .location | default "MESH_EXTERNAL" }}
  resolution: {{ .resolution | default "NONE" }}
---
{{- end -}}

{{- range .Values.destinationRules }}
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: {{ .name }}
  namespace: {{ .namespace | default $.Release.Namespace }}
spec:
  host: "{{ .host }}"
  trafficPolicy:
    tls:
      mode: {{ .mode | default "MUTUAL" }}
      {{ if ne .mode "DISABLE" }}
      clientCertificate: /etc/certs/cert-chain.pem
      privateKey: /etc/certs/key.pem
      caCertificates: /etc/certs/root-cert.pem
      {{ end }}
    {{- if .sbitimers }}
    connectionPool:
      tcp:
        connectTimeout: {{ .tcpConnectTimeout | default "750ms" }}
        tcpKeepalive:
          probes: {{ .tcpKeepAliveProbes | default 3 }}
          time: {{ .tcpKeepAliveTime | default "1500ms" }}
          interval: {{ .tcpKeepAliveInterval | default "1s" }}
    {{- end }}
---
{{- end -}}

{{- range .Values.envoyFilters_v_16x }}
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: {{ .name }}
  namespace: {{ .namespace | default $.Release.Namespace }}
spec:
  workloadSelector:
    labels:
      {{ .labelselector }}
  configPatches:
  {{- range .configpatch }}
  - applyTo: {{ .applyTo | default "NETWORK_FILTER" }}
```

```
      match:
        {{- if .patchContext }}
        context: {{ .patchContext }}
        {{- end }}
        {{- if eq .applyTo "CLUSTER" }}
        cluster:
          service:  {{ .clusterservice }}
        {{- end }}
        {{- if eq .applyTo "HTTP_ROUTE" }}
        routeConfiguration:
          {{- if .httpRoute_routeConfiguration_port }}
          portNumber: {{ .httpRoute_routeConfiguration_port }}
          {{- end }}
          vhost:
            name: {{ required ( printf "Error in %s envoyFilter. vhostname is
required for routeConfiguration when applyTo is
HTTP_ROUTE." .name ) .vhostname }}
        {{- else if eq .applyTo "FILTER_CHAIN" }}
        listener:
          {{- if .filterChain_listener_port }}
          portNumber: {{ .filterChain_listener_port }}
          {{- end }}
        {{- else if eq .applyTo "NETWORK_FILTER" }}
        listener:
          {{- if .filtername }}
          filterChain:
            filter:
              name: {{ .filtername | default
"envoy.http_connection_manager" }}
          {{- end }}
          {{- if .networkFilter_listener_port }}
          portNumber: {{ .networkFilter_listener_port }}
          {{- end }}
        {{- end }}
      patch:
        operation: {{ .operation | default "MERGE" }}
        value:
          {{- if .typeconfig }}
          typed_config:
            '@type': {{ .typeconfig }}
            {{ .configkey }}: {{ .configvalue }}
            {{- if .stream_idle_timeout }}
            stream_idle_timeout: {{ .stream_idle_timeout }}
            {{- end }}
            {{- if .max_stream_duration }}
            common_http_protocol_options:
              max_stream_duration: {{ .max_stream_duration }}
            {{- end }}
          {{- end }}
          {{- if .transport_socket_connect_timeout }}
          transport_socket_connect_timeout:
{{ .transport_socket_connect_timeout }}
          {{- end }}
          {{- if or .route_idle_timeout .route_max_stream_duration }}
          route:
            {{- if .route_idle_timeout }}
```

```
            idle_timeout: {{ .route_idle_timeout }}
            {{- end }}
            {{- if .route_max_stream_duration }}
            max_stream_duration:
              max_stream_duration: {{ .route_max_stream_duration }}
            {{- end }}
          {{- end }}
          {{- if .logicaldns }}
          type: {{ .logicaldns }}
          {{- end }}
    {{- end -}}
  ---
  {{- end -}}

  {{- range .Values.envoyFilters_v_19x_111x }}
  apiVersion: networking.istio.io/v1alpha3
  kind: EnvoyFilter
  metadata:
    name: {{ .name }}
    namespace: {{ .namespace | default $.Release.Namespace }}
  spec:
    workloadSelector:
      labels:
        {{ .labelselector }}
    configPatches:
    {{- range .configpatch }}
    - applyTo: {{ .applyTo | default "NETWORK_FILTER" }}
      match:
        {{- if .patchContext }}
        context: {{ .patchContext }}
        {{- end }}
        {{- if eq .applyTo "CLUSTER" }}
        cluster:
          service:  {{ .clusterservice }}
        {{- end }}
        {{- if eq .applyTo "HTTP_ROUTE" }}
        routeConfiguration:
          {{- if .httpRoute_routeConfiguration_port }}
          portNumber: {{ .httpRoute_routeConfiguration_port }}
          {{- end }}
          vhost:
            name: {{ required ( printf "Error in %s envoyFilter. vhostname is
required for routeConfiguration when applyTo is
HTTP_ROUTE." .name ) .vhostname }}
        {{- else if eq .applyTo "FILTER_CHAIN" }}
        listener:
          {{- if .filterChain_listener_port }}
          portNumber: {{ .filterChain_listener_port }}
          {{- end }}
        {{- else if eq .applyTo "LISTENER" }}
        listener:
          {{- if .listener_port }}
          portNumber: {{ .listener_port }}
          {{- end }}
        {{- else if eq .applyTo "NETWORK_FILTER" }}
        listener:
```

```
          {{- if .filtername }}
          filterChain:
            filter:
              name: {{ .filtername | default
"envoy.http_connection_manager" }}
          {{- end }}
          {{- if .networkFilter_listener_port }}
          portNumber: {{ .networkFilter_listener_port }}
          {{- end }}
        {{- end }}
      patch:
        operation: {{ .operation | default "MERGE" }}
        value:
          {{- if .typeconfig }}
          typed_config:
            '@type': {{ .typeconfig }}
            {{ .configkey }}: {{ .configvalue }}
            {{- if .stream_idle_timeout }}
            stream_idle_timeout: {{ .stream_idle_timeout }}
            {{- end }}
            {{- if .max_stream_duration }}
            common_http_protocol_options:
              max_stream_duration: {{ .max_stream_duration }}
            {{- end }}
          {{- end }}
          {{- if .transport_socket_connect_timeout }}
          transport_socket_connect_timeout:
{{ .transport_socket_connect_timeout }}
          {{- end }}
          {{- if or .route_idle_timeout .route_max_stream_duration }}
          route:
            {{- if .route_idle_timeout }}
            idle_timeout: {{ .route_idle_timeout }}
            {{- end }}
            {{- if .route_max_stream_duration }}
            max_stream_duration:
              max_stream_duration: {{ .route_max_stream_duration }}
            {{- end }}
          {{- end }}
          {{- if .logicaldns }}
          type: {{ .logicaldns }}
          {{- end }}
          {{- if .exactbalance }}
          connection_balance_config:
            exact_balance: {}
          {{- end }}
    {{- end }}
  ---
  {{- end -}}

  {{- range .Values.peerAuthentication }}
  apiVersion: security.istio.io/v1beta1
  kind: PeerAuthentication
  metadata:
    name: {{ .name }}
    namespace: {{ .namespace | default $.Release.Namespace }}
```

```
spec:
  {{ if .labelselector }}
  selector:
     matchLabels:
      {{ .labelselector }}
  {{ end }}
  mtls:
    mode: {{ .tlsmode }}
---
{{- end -}}

{{- range .Values.virtualService }}
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: {{ .name }}
  namespace: {{ .namespace | default $.Release.Namespace }}
spec:
  hosts:
  - {{ .host }}
  exportTo: {{ .exportTo }}
  http:
  - route:
    - destination:
        host: {{ .destinationhost }}
        port:
          number: {{ .port }}
    {{- if .timeout }}
    timeout: {{ .timeout }}
    {{- end }}
    {{ if or .attempts .retryon  }}
    retries:
          {{ if .attempts }}
             attempts: {{ .attempts }}
          {{ end }}
          {{ if .retryon }}
             retryOn: {{ .retryon }}
          {{ end }}
    {{ end }}
---
{{- end -}}

{{- range .Values.requestAuthentication }}
apiVersion: security.istio.io/v1beta1
kind: RequestAuthentication
metadata:
  name: {{ .name }}
  namespace: {{ .namespace | default $.Release.Namespace }}
spec:
  selector:
    matchLabels:
      {{ .labelselector }}
  jwtRules:
  - issuer: {{ .issuer }}
    {{ if .jwksUri }}
    jwksUri: {{ .jwksUri }}
```

```
    {{ end }}
    {{ if .jwks }}
    jwks: {{ .jwks }}
    {{ end }}
    forwardOriginalToken: true
---
{{- end -}}

{{- range .Values.authorizationPolicies }}
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: {{ .name }}
  namespace: {{ .namespace | default $.Release.Namespace }}
spec:
  action: {{ .action }}
  rules:
  - to:
    - operation:
        {{- if .hosts }}
        hosts:
        {{- range .hosts }}
        - {{ . | quote -}}
        {{- end }}
        {{- end }}
        {{- if .paths }}
        paths:
        {{- range .paths }}
        - {{ . | quote -}}
        {{- end }}
        {{- end }}
    {{- if .xfccvalues }}
    when:
    - key: request.headers[X-Forwarded-Client-Cert]
      values:
      {{- range .xfccvalues }}
      - {{ . | quote -}}
      {{- end }}
    {{- end }}
  selector:
    matchLabels:
      {{ .labelselector }}
---
{{- end -}}
```

Supported Fields in Custom Resource Definitions

Table 2-19 Supported Fields in Custom Resource Definitions

Custom Resource Definition	Fields
Service Entry	hosts
	exportTo
	addresses

Table 2-19 (Cont.) Supported Fields in Custom Resource Definitions

Custom Resource Definition	Fields
	ports.name
	ports.number
	ports.protocol
	resolution
Destination Rule	host
	mode
	sbitimers
	tcpConnectTimeout
	tcpKeepAliveProbes
	tcpKeepAliveTime
	tcpKeepAliveInterval
Envoy Filter	labelselector
	applyTo
	filtername
	operation
	typeconfig
	configkey
	configvalue
	stream_idle_timeout
	max_stream_duration
	patchContext
	networkFilter_listener_port
	transport_socket_connect_timeout
	filterChain_listener_port
	route_idle_timeout
	route_max_stream_duration
	httpRoute_routeConfiguration_port
	vhostname
	cluster.service
	type
	listener_port
	exactbalance
Peer Authentication	labelselector
	tlsmode
Virtual Service	host
	destinationhost
	port

Table 2-19 (Cont.) Supported Fields in Custom Resource Definitions

Custom Resource Definition	Fields
	exportTo
	retryon
	attempts
	timeout
Request Authentication	labelselector
	issuer
	jwks/jwksUri
Policy Authorization	labelselector
	action
	hosts
	paths
	xfccvalues

2.2.1.14.2 Installation of ASM Configuration Charts

Perform the below steps to install ASM configuration charts:

1. Download the ASM charts.tar file `ocudr-servicemesh-config-24.2.0.tgz` available with the CSAR files. .
2. Configure the custom-values.yaml file as follows:
   - Modify only the data section for all the Custom Resource Definition (CRD). You can comment the CRDs that you do not need. For example, in the CRD **Service Entry** modify the default values for all the fields with the values that you want to apply. This is applicable to all the supported CRDs.
   - The key and value mentioned for each CRD's is replaced in the template section. For example, the following sample envoy filters data shows how to configure the sample envoy filters using different FILTERS, PORTS:

```
envoyFilters_v_19x_111x:
  - name: xfccfilter
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnecti
onManager
        configkey: forward_client_cert_details
        configvalue: ALWAYS_FORWARD_ONLY
  - name: serverheaderfilter
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
```

```
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnecti
onManager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: custom-http-stream
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnecti
onManager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
        stream_idle_timeout: "6000ms"
        max_stream_duration: "7000ms"
        patchContext: SIDECAR_OUTBOUND
        networkFilter_listener_port: 8000
  - name: custom-tcpsocket-timeout
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: FILTER_CHAIN
        patchContext: SIDECAR_INBOUND
        operation: MERGE
        transport_socket_connect_timeout: "750ms"
        filterChain_listener_port: 8000
  - name: custom-http-route
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: HTTP_ROUTE
        patchContext: SIDECAR_OUTBOUND
        operation: MERGE
        route_idle_timeout: "6000ms"
        route_max_stream_duration: "7000ms"
        httpRoute_routeConfiguration_port: 8000
        vhostname: "ocudr.svc.cluster:8000"
  - name: logicaldnscluster
    labelselector: "app.kubernetes.io/instance: ocudr"
    configpatch:
      - applyTo: CLUSTER
        clusterservice: rchltxekvzwcamf-y-ec-
x-002.amf.5gc.mnc480.mcc311.3gppnetwork.org
        operation: MERGE
        logicaldns: LOGICAL_DNS
      - applyTo: CLUSTER
        clusterservice: rchltxekvzwcamd-y-ec-
x-002.amf.5gc.mnc480.mcc311.3gppnetwork.org
        operation: MERGE
        logicaldns: LOGICAL_DNS
  - name: slfexactbalancesample
    labelselector: "app.kubernetes.io/name: ingressgateway-prov"
    configpatch:
      - applyTo: LISTENER
        patchContext: SIDECAR_OUTBOUND
```

```
listener_port: 5001
operation: MERGE
exactbalance: true
```

3. Install the ASM configuration charts as below:
   - Run the below Helm install command to the namespace you want to apply the changes:

     ```
     helm install <Release-Name> -n <Namespace>
     ```

   - Run the below command to verify if all CRDs are installed:

     ```
     kubectl get <CRD-Name> -n <Namespace>
     ```

     ```
     [cloud-user@cne-cicd-23-1-bastion-1 ocudr-servicemesh-config]$ kubectl get se -n ocudr
     NAME              HOSTS                                        LOCATION        RESOLUTION   AGE
     kube-api-server   ["kubernetes.default.svc.cluster.local"]     MESH_INTERNAL   NONE         8d
     nrf               ["nrf123.oracle.com"]                        MESH_EXTERNAL   NONE         8d
     ```

   - Run the below Helm upgrade command to make changes in the custom-values.yaml file:

     ```
     helm upgrade <Release-Name> -n <Namespace>
     ```

2.2.1.14.3 Verification of Charts

Pre-requisite: A system has service mesh deployed with few ASM configurations like service entries and destination rule for UDR or SLF communication to NRF. Example: nrf1.abc.com.

To update and verify the service mesh configurations:

1. Unzip the attached zip file which contains ocudr asm charts.
2. Use the ocudr-asm-config's values.yaml file to add a new SE and DR with nrf1.abc.com and nrf2.abc.com.
3. Modify the asmResourceTemplate as needed.
4. Runt the helm3 upgrade command on the values.yaml file.
5. Use the `kubectl get se` and `kubectl get dr` commands to verify the updates and ensure service mesh applies these new changes.

For Inter-Cluster NF Communication Service Mesh verification, perform the following steps:

1. For Source NF configuration, add the destination rule mode as 'DISABLE' and in service entry, provide the host (which is configured on DNS) and load balancer port details.

   ```
   destinationRules:
      - hosts: nrf124.oracle.com
        mode: DISABLE
        name: nrf
   ```

2. For destination NF, configure the peerAuthentication for the namespace with mTLS mode as 'PERMISSIVE'. A sample template is as follows:

Peer Authentication:

```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: allservice
  namespace: ocudr
spec:
  mtls:
    mode: PERMISSIVE
```

3. After performing the above steps, verify the inter cluster communication between two NF's.

2.2.1.14.4 ASM Specific Configuration

To configure ASM

- Add the following annotation under global section:

```
# ********  Sub-Section Start: Custom Extension Global Parameters ********

#************************************************************************
global:
  customExtension:
    allResources:
      labels: {}
      annotations:
        oracle.com/cnc: "true"

    lbServices:
      labels: {}
      annotations: {}

    lbDeployments:
      labels: {}
      annotations: {}

    nonlbServices:
      labels: {}
      annotations: {}

    nonlbDeployments:
      labels: {}
      annotations: {}

  # ********  Sub-Section End: Custiom Extensions Global Parameters
********

#************************************************************************
```

- Enable the following flag under global section in custom values file:

```
global:

  # Servicemesh related configurations
```

```
# Enable when deployed with ASM
  serviceMeshCheck: true
```

- Enable the following flag under ingressgateway section.

```
ingressgateway:

  global:

    # In case of ASPEN Service Mesh enabled, to support clear text traffic
from outside of the cluster below flag needs to be true.

    istioIngressTlsSupport:

      ingressGateway: true
```

- Change Ingress Gateway Service Type to ClusterIP under the ingressgateway section.

```
ingressgateway:
  global:
    # Service Type
    type: ClusterIP
```

- Exclude actuator ports from Aspen mesh to avoid traffic through side car. These ports should be same as actuator ports (used for readiness/liveness checks) for Ingress Gateway and UDR microservices.
  The default actuator port (service.port.management) used for UDR microservices is 9000 and ingress/egress gateway is 9090 (ingressgateway.ports.actuatorPort). If the default ports are not changed then use the same annotation as given below. Also, exclude the management port of nudr-nrf-client-service from nudr-nrf-client-service's side car.

```
nudr-nrf-client-service:
  deployment:
    customExtension:
      labels: {}
      annotations:
        traffic.sidecar.istio.io/excludeInboundPorts: "9000"
```

> **Note**
>
> Exclude the management port of nudr-nrf-client-service from nudr-nrf-client-service side car.

- This step is conditional. In custom-values.yaml file, if any port is listed with **excludeInboundPorts** and **excludeOutboundPorts** annotation for ingress gateway, egressgateway, and alternate-route services, then you are required to exclude the coherence ports along with the port added. Following are the ports used for coherence by default.

```
traffic.sidecar.istio.io/excludeInboundPorts: 8095, 8096 9090, 7, 53
traffic.sidecar.istio.io/excludeOutboundPorts: 8095, 8096 9090, 7, 53
```

- Create a destination rule and service entry to enable MYSQL connectivity service to establish a connection between UDR/SLF and NDB cluster. This is outside ASM. Sample template is given below:
  Creating a Service for External MySQL Instance

```
apiVersion: v1
kind: Endpoints
metadata:
  name: mysql-connectivity-service-headless
  namespace: <ocudr-namespace>
subsets:
- addresses:
  - ip: <sql-node1-ip>
  - ip: <sql-node2-ip>
  ports:
  - port: 3306
    protocol: TCP
---
apiVersion: v1
kind: Service
metadata:
  name: mysql-connectivity-service-headless
  namespace: <ocudr-namespace>
spec:
  clusterIP: None
  ports:
  - port: 3306
    protocol: TCP
    targetPort: 3306
  sessionAffinity: None
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  name: mysql-connectivity-service
  namespace: <ocudr-namespace>
spec:
  externalName: mysql-connectivity-service-headless.<ocudr-
namespace>.svc.cluster.local
  sessionAffinity: None
  type: ExternalName
```

  Creating Service Entry and Destination Rule for External Database Instance

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: mysql-external-se
  namespace: <ocudr-namespace>
spec:
  hosts:
  - mysql-connectivity-service-headless.<ocudr-namespace>.svc.cluster.local
  ports:
  - number: 3306
```

```
    name: mysql
    protocol: MySQL
  location: MESH_EXTERNAL
---
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: mysql-external-dr
  namespace: <ocudr-namespace>
spec:
  host: mysql-connectivity-service-headless.<ocudr-
namespace>.svc.cluster.local
  trafficPolicy:
    tls:
      mode: DISABLE
```

> **Note**
>
> Do not use this template if you are using cnDBTier with ASM as backend for UDR/SLF 1.9.0 or above.

- If you enable **nudr-ondemand-migration** service, then a policy for 4G UDR connectivity is required as it is outside ASM.
  A sample template to create an external Kubernetes service for 4G UDR along with service entry and destination rule is given below:

  External Kubernetes Service for 4G UDR

```
apiVersion: v1
kind: Endpoints
metadata:
  name: sourceudr-service-headless
  namespace: <ocudr-namespace>
subsets:
- addresses:
  - ip: <4gudr-destination-ip>
  ports:
  - port: <4gudr-destination-port>
    protocol: TCP
---
apiVersion: v1
kind: Service
metadata:
  name: sourceudr-service-headless
  namespace: <ocudr-namespace>
spec:
  clusterIP: None
  ports:
  - port: <4gudr-destination-port>
    protocol: TCP
    targetPort: <4gudr-destination-port>
  sessionAffinity: None
  type: ClusterIP
---
apiVersion: v1
```

```
kind: Service
metadata:
  name: sourceudr-service
  namespace: <ocudr-namespace>
spec:
  externalName: sourceudr-service-headless.<ocudr-
namespace>.svc.cluster.local
  sessionAffinity: None
  type: ExternalName
```

Service Entry and Destination Rule (for the service created in the previous template)

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: sourceudr-external-se
  namespace: <ocudr-namespace>
spec:
  hosts:
  - sourceudr-service-headless.<ocudr-namespace>.svc.cluster.local
  ports:
  - number: <4gudr-destination-port>
    name: sourceudr
  location: MESH_EXTERNAL
---
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: sourceudr-external-dr
  namespace: <ocudr-namespace>
spec:
  host: sourceudr-service-headless.<ocudr-namespace>.svc.cluster.local
  trafficPolicy:
    tls:
      mode: DISABLE
```

- If you are using nudr-migration tool with ASM setup, it is mandatory to create a service entry and a destination rule for 4G UDR IP along with the port used, which is outside ASM. Use same IP Address given in the host IP details of nudr-migration template. A sample template is given below:

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: sourceudr-external-se
  namespace: <ocudr-namespace>
spec:
  hosts:
  - sourceudr-service-headless.<ocudr-namespace>.svc.cluster.local
  addresses:
  - 10.196.7.251
  ports:
  - number: 3867
    name: firstconnectionport
  - number: 3866
```

```
    name: secondconnectionport
  location: MESH_EXTERNAL---
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: sourceudr-external-dr
  namespace: <ocudr-namespace>
spec:
  host: sourceudr-service-headless.<ocudr-namespace>.svc.cluster.local
  trafficPolicy:
    tls:
      mode: DISABLE
```

- To run XFCC and Server header related ATS test cases on SLF or UDR with ASM, envoy filters are used to forward the header passed by client. Sample envoy filters template is given below:

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: <server-header envoyfilter name>
  namespace: <NF deployment namespace>
spec:
  workloadSelector:
    labels:
      app.kubernetes.io/instance: <helm release name>
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      listener:
        filterChain:
          filter:
            name: "envoy.http_connection_manager"
    patch:
      operation: MERGE
      value:
        typed_config:
          '@type': type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManage
r
          server_header_transformation: PASS_THROUGH

---
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: <xfcc envoyfilter name>
  namespace: <NF deployment namespace>
spec:
  workloadSelector:
    labels:
      app.kubernetes.io/instance: <helm release name>
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      listener:
```

```
          filterChain:
            filter:
              name: "envoy.http_connection_manager"
      patch:
        operation: MERGE
        value:
          typed_config:
            '@type': type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManage
r
            forward_client_cert_details: ALWAYS_FORWARD_ONLY
```

- For inter pod communication on UDR or SLF deployed with ASM, create **Peer Authentication** on pods. Sample template is given below:
  Peer Authentication

```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: <Peer Authentication name>
  namespace: <UDR/SLF deployment namespace>
spec:
  selector:
    matchLabels:
      app.kubernetes.io/part-of: ocudr
  mtls:
    mode: PERMISSIVE
```

2.2.1.14.5 Deleting ASM

This section describes the steps to delete ASM.

To delete ASM, run the following command:

```
helm delete <helm-release-name> -n <namespace>
```

Where,

- `<helm-release-name>` is the release name used by the helm command. This release name must be the same as the release name used for ServiceMesh.
- `<namespace>` is the deployment namespcae used by Helm command.

For example:

```
helm delete ocudr-servicemesh-config -n ocudr
```

To disable ASM, run the following command:

```
kubectl label --overwrite namespace ocudr istio-injection=disabled
```

To verify if ASM is disabled, run the following command:

```
kubectl get se,dr,peerauthentication,envoyfilter,vs -n ocudr
```

2.2.1.15 Configuring udrServices and consumerNF Flags

Before installing or upgrading UDR, it is important to configure udrServices and consumerNF flags.

- udrService: This flag indicates UDR mode. It can be either All, nudr-dr, or nudr-group-id-map (slf). On the basis of udrService value, NF profiles regiater with NRF. If the udrService value is 'All' or 'nudr-dr', it is filtered further on the basis of consumerNF flag.
- consumerNF: This flag indicates consumer network functions for UDR. It is a comma separated list of possible consumer services like PCF, UDM, NEF, or any combination of these services.

To configure the udrService and consumerNF flags, use the following information.

- If the udrService is All, then on the basis of udrMode and consumerNF type, set the value of allowedNFTypes in appProfile.

> **Note**
>
> NRF needs to be added in allowedNFType.

The default value of consumerNF type in this mode is PCF, UDM, NEF. The details are:

Table 2-20 Allowed NF Type Values When udrService is All

Consumer NF	Value of Allowed NF Type (a field in App PROFILE)	Notes
PCF	"allowedNfTypes":["PCF","NRF"]	Only PCF related entries are loaded in udrdb.
UDM	"allowedNfTypes":["UDM","NRF"]	Only UDM related entries are loaded in udrdb. Make sure diameter proxy and diameter gateway microservices are disabled.
NEF	"allowedNfTypes":["NEF","NRF"]	Only NEF related entries are loaded in udrdb. Make sure diameter proxy and diameter gateway microservices are disabled.
PCF, UDM	"allowedNfTypes": ["PCF","NRF","UDM"]	Only PCF and UDM related entries are loaded in udrdb.
PCF, NEF	"allowedNfTypes": ["PCF","NRF","NEF"]	Only PCF,NEF related entries are loaded in udrdb.
UDM, NEF	"allowedNfTypes": ["UDM","NRF","NEF"]	Only UDM and NEF related entries are loaded in udrdb. Make sure diameter proxy and diameter gateway microservices are disabled.
PCF, UDM, NEF (default)	"allowedNfTypes": ["PCF","UDM","NRF","NEF"]	PCF, UDR, and UDM related entries are loaded in udrdb

The default app profile request is:


```
[{"nfInstanceId":"5a7bd676-ceeb-44bb-95e0-
f6a55a328b03","nfStatus":"REGISTERED","fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","nfType":"UDR","allowedNfTypes":
["PCF","UDM","NRF"],"plmnList":
[{"mnc":"14","mcc":"310"}],"priority":10,"capacity":500,"load":0,"locality"
:"bangalore","nfServices":[{"load":0,"scheme":"http","versions":
[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","ipEndPoints":
[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus
":"REGISTERED","allowedNfTypes":
["PCF","UDM"],"serviceInstanceId":"ae870316-384d-458a-
bd45-025c9e748976","serviceName":"nudr-dr","priority":10,"capacity":500},
{"load":0,"scheme":"http","versions":
[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","ipEndPoints":
[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus
":"REGISTERED","allowedNfTypes":
["NRF"],"serviceInstanceId":"547d42af-628a-4d5d-
a8bd-38c4ba672682","serviceName":"nudr-group-id-
map","priority":10,"capacity":500}],"udrInfo":{"supportedDataSets":
["POLICY","SUBSCRIPTION"],"groupId":"udr-1","externalGroupIdentifiersRanges
":[{"start":"10000000000","end":"20000000000"}],"supiRanges":
[{"start":"10000000000","end":"20000000000"}],"gpsiRanges":
[{"start":"10000000000","end":"20000000000"}]},"heartBeatTimer":90,"nfServi
cePersi
```


- If the udrService is nudr-dr, then on the basis of udrMode and consumerNF type, set the value of allowedNFTypes in appProfile. The default value of consumerNF is PCF.

Table 2-21 Allowed NF Type Values When udrService is nudr-dr

Consumer NF	Value of Allowed NF Type (a field in NF PROFILE)	Notes
PCF (default)	"allowedNfTypes":["PCF"]	Only PCF related entries are loaded in udrdb.
UDM	"allowedNfTypes":["UDM"]	Only UDM related entries are loaded in udrdb. Make sure diameter proxy and diameter gateway microservices are disabled.
NEF	"allowedNfTypes":["NEF"]	Only NEF related entries are loaded in udrdb. Make sure diameter proxy and diameter gateway microservices are disabled.
PCF, UDM	"allowedNfTypes":["PCF","UDM"]	Only PCF and UDM related entries are loaded in udrdb.
PCF, NEF	"allowedNfTypes":["PCF","NEF"]	Only PCF and NEF related entries are loaded in udrdb.

Table 2-21 (Cont.) Allowed NF Type Values When udrService is nudr-dr

Consumer NF	Value of Allowed NF Type (a field in NF PROFILE)	Notes
UDM, NEF	"allowedNfTypes":["UDM","NEF"]	Only UDM and NEF related entries are loaded in udrdb. Make sure diameter proxy and diameter gateway microservices are disabled.
PCF, UDM, NEF	"allowedNfTypes": ["PCF","UDM","NEF"]	PCF, UDR, and UDM related entries are loaded in udrdb.

The default appProfile request is:

```
nudr-dr [{"nfInstanceId":"5a7bd676-ceeb-44bb-95e0-
f6a55a328b03","nfStatus":"REGISTERED","fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","nfType":"UDR","allowedNfTypes":
["PCF","UDM","NRF"],"plmnList":
[{"mnc":"14","mcc":"310"}],"priority":10,"capacity":500,"load":0,"locality"
:"bangalore","nfServices":[{"load":0,"scheme":"http","versions":
[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","ipEndPoints":
[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus
":"REGISTERED","allowedNfTypes":
["PCF","UDM"],"serviceInstanceId":"ae870316-384d-458a-
bd45-025c9e748976","serviceName":"nudr-
dr","priority":10,"capacity":500}],"udrInfo":{"supportedDataSets":
["POLICY","SUBSCRIPTION"],"groupId":"udr-1","externalGroupIdentifiersRanges
":[{"start":"10000000000","end":"20000000000"}],"supiRanges":
[{"start":"10000000000","end":"20000000000"}],"gpsiRanges":
[{"start":"10000000000","end":"20000000000"}]},"heartBeatTimer":90,"nfServi
cePersistence":false,"nfProfileChangesSupportInd":false,"nfSetIdList":
["setxyz.udrset.5gc.mnc012.mcc345"]}]
```

- If the udrService is nudr-group-id-map, then set the allowedNFType in appProfile as NRF. The default appProfile request is:

```
[{"nfInstanceId":"5a7bd676-ceeb-44bb-95e0-
f6a55a328b03","nfStatus":"REGISTERED","fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","nfType":"UDR","allowedNfTypes":
["NRF"],"plmnList":
[{"mnc":"14","mcc":"310"}],"priority":10,"capacity":500,"load":0,"locality"
:"bangalore","nfServices":[{"load":0,"scheme":"http","versions":
[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-
ingressgateway.myudr.svc.cluster.local","ipEndPoints":
[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus
":"REGISTERED","allowedNfTypes":
["NRF"],"serviceInstanceId":"547d42af-628a-4d5d-
a8bd-38c4ba672682","serviceName":"nudr-group-id-
map","priority":10,"capacity":500}],"udrInfo":
{"groupId":"udr-1","externalGroupIdentifiersRanges":
[{"start":"10000000000","end":"20000000000"}],"supiRanges":
[{"start":"10000000000","end":"20000000000"}],"gpsiRanges":
[{"start":"10000000000","end":"20000000000"}]},"heartBeatTimer":90,"nfServi
```

```
cePersistence":false,"nfProfileChangesSupportInd":false,"nfSetIdList":
["setxyz.udrset.5gc.mnc012.mcc345"]}]
```

2.2.1.16 Configuring eirServices and consumerNF Flags

Before installing or upgrading EIR, it is important to configure udrServices and consumerNF flags.

- udrService: eirService indicates EIR mode. eirService is n5g-eir-eic. Set the parameter global.udrServices to n5g-eir-eic to indicate EIR mode. n5g-eir-eic is the service used to register EIR with NRF.
- consumerNF: This flag indicates consumer network functions for EIR.

To configure the eirService and consumerNF flags, use the following information.

- If the eirService is n5g-eir-eic, then on the basis of eirMode and consumerNF type, set the value of allowedNfTypes in appProfile. The default value of consumerNF is AMF.

> **Note**
>
> NRF needs to be added in allowedNFType.

Table 2-22 Allowed NF Type Values When eirService is n5g-eir-eic

Consumer NF	Value of Allowed NF Type (a field in App PROFILE)	Notes
AMF	"allowedNfTypes":["AMF"]	Only AMF related entries are loaded.

The default app profile request is:

```
[{
    "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
    "nfStatus": "REGISTERED",
    "fqdn": "ocudr-ingressgateway-sig.myudr.svc.cluster.local",
    "nfType": "5G_EIR",
    "allowedNfTypes": ["AMF"],
    "plmnList": [{
        "mnc": "14",
        "mcc": "310"
    }],
    "priority": 10,
    "capacity": 500,
    "load": 0,
    "locality": "bangalore",
    "nfServices": [{
        "load": 0,
        "scheme": "http",
        "versions": [{
            "apiFullVersion": "2.1.0.alpha-3",
            "apiVersionInUri": "v1"
        }],
        "fqdn": "ocudr-ingressgateway-sig.myudr.svc.cluster.local",
        "ipEndPoints": [{
```

```
                "port": "80",
                "ipv4Address": "10.0.0.0",
                "transport": "TCP"
            }],
            "nfServiceStatus": "REGISTERED",
            "allowedNfTypes": ["AMF"],
            "serviceInstanceId": "ae870316-384d-458a-bd45-025c9e748976",
            "serviceName": "n5g-eir-eic",
            "priority": 10,
            "capacity": 500
        }],
        "heartBeatTimer": 90,
        "nfServicePersistence": false,
        "nfProfileChangesSupportInd": false,
        "nfSetIdList": ["setxyz.udrset.5gc.mnc012.mcc345"]
}]
```

2.2.1.17 Configuring PodDisruptionBudget Kubernetes Resource

The PodDisruptionBudget (PDB) is a Kubernetes resource, which you can use for the nudr-drservice, nudr-notify-service, nudr-config service, config-server, nudr-diameter-proxy, and nudr-diamgateway services. This PDB resource allows you to specify the number of concurrent disruptions for your pods. It is helpful when a pod is deleted either deliberately or accidentally.

> **Note**
>
> The performance and capacity of the UDR system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment, including but not limited to, the size of the json payload, operation type, and traffic model.

When a Kubernetes node runs out of RAM or disk space, the pod evicts in an attempt to ensure the node is running properly. It uses the **maxAvailable** parameter specified in the Helm chart to determine the minimum number of pods that are available during a disruption, even in the absence of the evicted pod.

> **Note**
>
> The value of the **maxAvailable** parameter is an absolute number.

Table 2-23 PodDisruptionBudget for SLF Deployment

Microservice	PDB Value Specified (maxUnavailable)	Notes
ingressgateway	1	NA
egressgateway	1	NA
nudr-drservice	1	NA
nudr-dr-provservice	1	NA
nudr-config	1	NA
config-server	1	NA

Table 2-23 (Cont.) PodDisruptionBudget for SLF Deployment

Microservice	PDB Value Specified (maxUnavailable)	Notes
alternate-route	1	NA
nrf-client	1	NA
app-info	1	NA
perf-info	1	NA
perf-info-prov	1	NA

Table 2-24 PodDisruptionBudget for UDR Deployment

Microservice	PDB Value Specified (maxUnavailable)	Notes
ingressgateway	1	NA
egressgateway	1	NA
nudr-drservice	1	NA
nudr-dr-provservice	1	NA
nudr-notify-service	1	NA
nudr-diam-gateway	1	NA
nudr-diameterproxy	1	NA
nudr-ondemand-migration	1	NA
nudr-config	1	NA
config-server	1	NA
alternate-route	1	NA
nrf-client	1	NA
app-info	1	NA
perf-info	1	NA
perf-info-prov	1	NA

Table 2-25 PodDisruptionBudget for EIR Deployment

Microservice	PDB Value Specified (maxUnavailable)	Notes
ingressgateway	1	NA
egressgateway	1	NA
nudr-drservice	1	NA
nudr-dr-provservice	1	NA
nudr-config	1	NA
config-server	1	NA
alternate-route	1	NA
nrf-client	1	NA
app-info	1	NA
perf-info	1	NA
perf-info-prov	1	NA

2.2.1.18 Configuring Network Policies

Overview

The network policies allow ingress or egress rules to be defined based on Kubernetes resources such as Pod, Namespace, IP, and Ports. These rules are selected based on Kubernetes labels in the application. These network policies enforces access restrictions for all the applicable data flows except communication from Kubernetes node to pod for invoking container probe.

> **Note**
>
> Configuring network policies is an optional step. Based on the security requirements, network policies may or may not be configured.

For more information on the network policy, see https://kubernetes.io/docs/concepts/services-networking/network-policies/.

Configuring Network Policies

Following are the various operations that can be performed for network policies after UDR deployment.

Installing Network Policies

The Network Policies are part of ocudr Helm package and can be modified from ocudr_custom_values.yaml file. The entire specification section in the ocudr_custom_values.yaml file is taken as input for network policies. You can provide names to each policy to differentiate the policies when deployed. The template file for network policy is as below:

```
#Template for various network Policy
{{- if .Values.global.createNetworkPolicy }}
{{- range .Values.global.networkPolicy }}
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: {{ .metadata.name }}
spec:
  {{- .spec | toYaml | nindent 2 }}
---
{{- end -}}
{{- end }}
```

The ocudr_custom_values.yaml file consists of various network policies for UDR that you can use to update the network policies. If there are no policies in a namespace by default, then all ingress and egress traffic is allowed to and from the pods in that namespace.

```
 - - metadata:
      name: deny-ingress-all-udr
    spec:
      podSelector:
        matchLabels:
```

```
        app.kubernetes.io/part-of: ocudr
    policyTypes:
    - Ingress
- metadata:
    name: allow-ingress-sbi-udr
  spec:
    podSelector:
      matchLabels:
        app.kubernetes.io/name: ingressgateway-sig
    policyTypes:
    - Ingress
    ingress:
    - {}

- metadata:
    name: allow-ingress-prov-udr
  spec:
    podSelector:
      matchLabels:
        app.kubernetes.io/name: ingressgateway-prov
    policyTypes:
    - Ingress
    ingress:
    - {}

- metadata:
    name: allow-ingress-diamgateway-udr
  spec:
    podSelector:
      matchLabels:
        app.kubernetes.io/name: nudr-diam-gateway
    policyTypes:
    - Ingress
    ingress:
    - ports:
      - protocol: TCP
        port: 3868

- metadata:
    name: deny-egress-all-ocudr
  spec:
    podSelector:
      matchLabels:
        app.kubernetes.io/part-of: ocudr
  policyTypes:
  - Egress

- metadata:
    name: allow-egress-through-egress-ocudr
  spec:
    podSelector:
      matchLabels:
        app.kubernetes.io/name: egressgateway
    policyTypes:
    - Egress
    egress:
```

```
      - {}

  - metadata:
      name: allow-egress-through-perf-ocudr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/name: perf-info
      policyTypes:
      - Egress
      egress:
      - {}

  - metadata:
      name: allow-egress-through-appinfo-ocudr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/name: appinfo
      policyTypes:
      - Egress
      egress:
      - {}

  - metadata:
      name: allow-egress-through-diamgateway-ocudr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/name: nudr-diam-gateway
      policyTypes:
      - Egress
      egress:
      - {}

# Allow ingress for Prometheus Metrics Scrapping
  - metadata:
      name: allow-ingress-prometheus-udr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Ingress
      ingress:
      - ports:
        - protocol: TCP
          port: 9090 # Prometheus Port
        - protocol: TCP
          port: 9000
        - protocol: TCP
          port: 9001
        - protocol: TCP
          port: 9002

  - metadata:
```

```
      name: allow-ingress-service-mesh-ocudr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Ingress
      ingress:
      - ports:
        - protocol: TCP
          port: 15021
        - protocol: TCP
          port: 15012
        - protocol: TCP
          port: 15000
        - protocol: TCP
          port: 15020

# Allow egress to DB SQL and Monitoring Ports
  - metadata:
      name: allow-mysql-db-udr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Egress
      egress:
      - ports:
        - protocol: TCP
          port: 3306 # DB SQL Port
        - protocol: TCP
          port: 8080 # DB Monitoring Port

# Allow egress to K8s API Port
  - metadata:
      name: allow-egress-k8s
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Egress
      egress:
      - ports:
        - protocol: TCP
          port: 6443 # K8s API Server Port

# Allow egress to Jaeger Agent
  - metadata:
      name: allow-jaeger-udr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
```

```
      - Egress
      egress:
      - ports:
        - protocol: TCP
          port: 4318 # Jaeger Agent Port

# Allow egress to DNS
  - metadata:
      name: allow-egress-dns-udr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Egress
      egress:
      - ports:
        - protocol: TCP
          port: 53   # DNS TCP Port
        - protocol: UDP
          port: 53   # DNS UDP Port

  - metadata:
       name: allow-egress-service-mesh-ocudr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Egress
      egress:
      - ports:
        - protocol: TCP
          port: 15021
        - protocol: TCP
          port: 15012
        - protocol: TCP
          port: 15000
        - protocol: TCP
          port: 15020

# Allow ingress from UDR Pods
  - metadata:
      name: allow-ingress-from-udr-pods
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Ingress
      ingress:
      - from:
        - podSelector:
            matchLabels:
              app.kubernetes.io/part-of: ocudr
```

```
# Allow egress to UDR Pods
  - metadata:
      name: allow-egress-to-udr-pods
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/part-of: ocudr
      policyTypes:
      - Egress
      egress:
      - to:
        - podSelector:
            matchLabels:
              app.kubernetes.io/part-of: ocudr

# Allow ingress from Console
  - metadata:
      name: allow-ingress-from-console-udr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/name: nudr-config
      policyTypes:
      - Ingress
      ingress:
       - from:
         - namespaceSelector: {}
           podSelector:
             matchLabels:
               app.kubernetes.io/part-of: cncc-core
         ports:
         - protocol: TCP
           port: 5001

  - metadata:
      name: allow-ingress-from-atspod-config-ocudr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/name: nudr-config
      policyTypes:
      - Ingress
      ingress:
      - from:
        - podSelector:
            matchLabels:
              app: ocats-udr
        ports:
        - protocol: TCP
          port: 5001
```

Network policies are applicable only to a particular namespace. If the policies are required to be valid across namespaces, for example, in CNC console, if a different namespace is installed than that of UDR then you must edit the policy for CNC console to nudr-config as follow:

```
- metadata:
    name: allow-ingress-from-console-udr
    spec:
      podSelector:
        matchLabels:
          app.kubernetes.io/name: nudr-config
      policyTypes:
      - Ingress
      ingress:
       - from:
         - namespaceSelector:
             matchLabels:
               kubernetes.io/metadata.name=<namespace of console>
           podSelector:
             matchLabels:
               app.kubernetes.io/part-of: cncc-core
         ports:
         - protocol: TCP
           port: 5001
```

When the remote transfer is enabled for subscriber export tool and subscriber bulk tool, you must uncomment the network policy section as follows to allow the transfer:

```
#Uncomment line no 76 to 88 If sftp is enabled and to transfer out the files
to remote server for export tool
  #- metadata:
  #    name: sftp-export-network-policy
  #    namespace: ocudr     # should be deployed ns
  #  spec:
  #    podSelector:
  #      matchLabels:
  #        app.kubernetes.io/name: nudr-export-tool
  #    policyTypes:
  #    - Egress
  #    egress:
  #    - to:
  #      - ipBlock:
  #          cidr: 10.121.29.66/32    #Give the remote server ip here

  #Uncomment line no 91 to 117 If sftp is enabled and to transfer out the
files to remote server for import tool
  #- metadata:
  #    name: sftp-import-egress-network-policy
  #    namespace: ocudr     # should be deployed ns
  #  spec:
  #    podSelector:
  #      matchLabels:
  #        app.kubernetes.io/name: nudr-bulk-import
  #    policyTypes:
  #    - Egress
  #    egress:
```

```
#    - to:
#      - ipBlock:
#          cidr: 10.75.229.137/32    #Give the remote server ip here

#- metadata:
#    name: sftp-import-ingress-network-policy
#    namespace: ocudr     # should be deployed ns
#  spec:
#    podSelector:
#      matchLabels:
#        app.kubernetes.io/name: nudr-bulk-import
#    policyTypes:
#    - Ingress
#    ingress:
#    - from:
#      - ipBlock:
#          cidr: 10.75.229.137/32    #Give the remote server ip here
```

2.2.1.18.1 Verifying Network Policies

Run the below command to check if network policies are deployed:

```
kubectl get networkpolicy -n <namespace>
```

The sample output of the network policy is as below:

```
allow-egress-dns-provgw                app.kubernetes.io/part-
of=provgw               4d3h
allow-egress-dns-udr                   app.kubernetes.io/part-
of=ocudr                7d19h
allow-egress-k8s                       app.kubernetes.io/part-
of=ocudr                7d19h
allow-egress-k8s-provgw                app.kubernetes.io/part-
of=provgw               4d3h
allow-egress-to-provgw-pods            app.kubernetes.io/part-
of=provgw               4d3h
allow-egress-to-udr-pods               app.kubernetes.io/part-
of=ocudr                7d19h
allow-ingress-from-console-provgw      app.kubernetes.io/name=provgw-
config           4d3h
allow-ingress-from-console-udr         app.kubernetes.io/name=nudr-
config             7d19h
allow-ingress-from-provgw-pods-provgw  app.kubernetes.io/part-
of=provgw               4d3h
allow-ingress-from-udr-pods            app.kubernetes.io/part-
of=ocudr                7d19h
allow-ingress-prometheus-provgw        app.kubernetes.io/part-
of=provgw               4d3h
allow-ingress-prometheus-udr           app.kubernetes.io/part-
of=ocudr                7d19h
allow-ingress-prov-udr                 app.kubernetes.io/name=ingressgateway-
prov     7d19h
allow-ingress-provgw                   app.kubernetes.io/name=prov-
ingressgateway     4d3h
```

```
allow-ingress-sbi-udr                    app.kubernetes.io/name=ingressgateway-
sig      7d19h
allow-jaeger-provgw                      app.kubernetes.io/part-
of=provgw               4d3h
allow-jaeger-udr                         app.kubernetes.io/part-
of=ocudr                7d19h
allow-mysql-db-provgw                    app.kubernetes.io/part-
of=provgw               4d3h
allow-mysql-db-udr                       app.kubernetes.io/part-
of=ocudr                7d19h
deny-egress-all-except-egw-provgw        app.kubernetes.io/name notin
(egressgateway)   4d3h
deny-egress-all-except-egw-udr           app.kubernetes.io/name notin
(egressgateway)   7d19h
deny-ingress-all-provgw                  app.kubernetes.io/part-
of=provgw               4d3h
deny-ingress-all-udr                     app.kubernetes.io/part-
of=ocudr                7d19h
```

Table 2-26 Supported Kubernetes Resource for Configuring Network Policies

Parameter	Description	Details
apiVersion	This indicates Kubernetes version for access control. **Note:** This is the supported API version for network policies. This is a read-only parameter.	**Data Type:** string **Default Value:** `networking.k8s.i o/v1`
kind	This represents the REST resource this object represents. **Note:** This is a read-only parameter.	**Data Type:** NetworkPolicy **Default Value:** `networking.k8s.i o/v1`
metadata.name	This indicates unique name for the network policy.	**Default Value:** `{{ .metadata.name }}` This is a mandatory parameter.
spec.{}	This consists of all the information needed to define a particular network policy in the given namespace.	**Default Value:** NA This is a mandatory parameter.

For more information about this functionality, see Network Policies in the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

2.2.1.19 Manual Creation of UDR Database and MySQL User

Follow the below steps to manaually create UDR Database and MySQL User:

1. Perform the following steps to create the UDR databases required for the deployment:
   a. Run the following command to log in to one of the ndbappmysqld node pods:

      ```
      kubectl exec -it ndbappmysqld-0 -n <db-namespace>
      ```

b. Once logged into the pod, run the following command with root credentials to login to mysql terminal:

```
mysql -h127.0.0.1 -uroot -p<rootPassword>
```

c. Run the following command to create the UDR subscriber and configuration database:

- `CREATE DATABASE IF NOT EXISTS <udrdbname> CHARACTER SET utf8;`
- `CREATE DATABASE IF NOT EXISTS <udrconfigdbname> CHARACTER SET utf8;`

> **Note**
>
> In case of multiple site deployment you can create the configuration database for all sites using the similar CREATE statements. This can be performed in one of the ndbappmysqld under any one site.

2. Perform the following steps to create the mysql user required for accessing the UDR databases. These instructions need to be executed on all ndbappmysqld and ndbmysql nodes:

a. Run the following command to log in to one of the sql node pods:

```
kubectl exec -it ndbappmysqld-x/ndbmysqld- -n <db-namespace>
```

b. Once logged in to the pod, run the following command with root credentials to login to mysql terminal:

```
mysql -h127.0.0.1 -uroot -p<rootPassword>
```

c. Run the following command to create mysql user and provide required GRANTS for the user:

- `CREATE USER IF NOT EXISTS '<udruser>'@'%' IDENTIFIED BY '<udrpasswd>';`
- `GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, CREATE TEMPORARY TABLES, DELETE, UPDATE,EXECUTE,REFERENCES ON <udrdbname>.* TO '<udruser>' @'%';`
- `GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, CREATE TEMPORARY TABLES, DELETE, UPDATE,EXECUTE,REFERENCES ON <udrconfigdbname>.* TO '<udruser>' @'%';`
- `FLUSH PRIVILEGES;`

> **Note**
>
> In case of multiple site deployment using the GRANT command provide access to all UDR configuration databases created in the Step 1.

2.2.1.20 Secondary Path for PCF Notifications When the Primary PCF is Unavailable

Follow the steps to use the Egress Gateway SBIRouting to perform notification request attempts to peer1 (secondary PCF or PCRF) when peer0 (primary PCF or PCRF) response is not successful.

> **Note**
>
> This configuration is specific for fixed hosts or IPs for PCF services.

Configure peerConfiguration and peerSetConfiguration from Egress Gateway sbiRouting Section

You can perform any of the following options:

> **Note**
>
> Irrespective of the options selected, you must perform the remaining procedures.

- **Option 1: Configure static hosts on Egress Gateway:** You can configure multiple PCF hosts as peers in the Egress Gateway Helm values as follows and this will be included to proxy whenever a request matched a routeConfig.
  - Set `sbiRerouteEnabled` to true and configure hosts in peerConfiguration and peerSetConfiguration with corresponding identifiers.
  - Refer the following sbiRouting yaml file reference to create peer0, peer1 peerConfiguration, and set0 peerSetConfiguration instances. Priority must be given to the peer with lowest value. Peer with priority 1 has higher priority than peer with priority 2.
  - The peer0, peer1, and set0 are custom identifiers used in the following example. If there are any change on the identifier name, you must modify all applicable strings in the custom-values.yaml.

```
egressgateway:
  sbiRouting:
    sbiRerouteEnabled: true
    peerConfiguration:
    - id: peer0
      host: <PCF1 SERVICE FQDN or IP>
      port: <SERVICE PORT>
      apiPrefix: "/"
    - id: peer1
      host: <PCF2 SERVICE FQDN or IP>
      port: <SERVICE PORT>
      apiPrefix: "/"
    peerSetConfiguration:
    - id: set0
      httpConfiguration:
      - priority: 1
        peerIdentifier: peer0
```

```
    - priority: 2
      peerIdentifier: peer1
```

- **Option 2: Configure static hosts on alternate route using a virtual FQDN:** You can configure a `virtualHost` in the Egress Gateway that uses a staticVirtualFqdns on the UDR alternate route. Since alternate route can resolve `virtualHost` and domains using Kubernetes cluster coreDNS. This options allows adding virtualFqnds/domains that were previously configured on the Kubernetes cluster coreDNS. You can decouple UDR configuration from FQDN or IP changes on the PCF side:
  - On the Egress Gateway, set the `sbiRerouteEnabled` to true and configure `virtualHost peerConfiguration` and `peerSetConfiguration` with corresponding identifiers.
  - On the alternate route, configure `staticVirtualFqdns` and set `dnsSrvEnabled` to true.
  - Refer the following sbiRouting yaml file reference to create peer0, peerConfiguration, set0 peerSetConfiguration and notify.test.com staticVirtualFqdns intances.
    * The virtualHost notify.test.com needs to match a configured staticVirtualFqdns name in alternate route.
    * The `pcf1.oracletest.com` and `pcf2.oracletest.com` are custom domains for the following example that needs to be previously configured on `coredns` service on the Kubernetes cluster.
    * Priority is given to the target with lowest value. Target with priority 10 has higher priority than target with priority 20. The peer0, notify.test.com, and set0 are custom identifiers used in the following example. If there are any change on the identifier name, you must modify all applicable strings in the custom-values.yaml.

```
egressgateway:
  sbiRouting:
    sbiRerouteEnabled: true
    peerConfiguration:
    - id: peer0
      virtualHost: notify.test.com
      apiPrefix: "/"
    peerSetConfiguration:
    - id: set0
      httpConfiguration:
      - priority: 1
        peerIdentifier: peer0

  # Configurations related to DNS SRV. Host and error details
  dnsSrv:
    host: 127.0.0.1
    alternateRouteSvcName: "alternate-route"
    port: 80
    scheme: http
    errorCodeOnDNSResolutionFailure: 425
    errorDescriptionOnDNSResolutionFailure: ""
    errorTitleOnDNSResolutionFailure: ""
    errorCauseOnDNSResolutionFailure: ""


alternate-route:
  staticVirtualFqdns:
    - name: http://notify.test.com
```

```
      alternateFqdns:
       - target: pcf1.oracletest.com # Domain configured on coreDNS for
    PCF1 SERVICE FQDN or IP
         port: 30195 # PCF1 Service port
         priority: 10
       - target: pcf2.oracletest.com # Domain configured on coreDNS for
    PCF2 SERVICE FQDN or IP
         port: 31775 # PCF2 Service port
         priority: 20

    #Flag to control if DNS-SRV queries are sent to coreDNS or not
      dnsSrvEnabled: true
```

Configure routesConfig

Set the `sbiRoutingEnabled` parameter as true in routesConfig identifier that corresponds to peerSetIdentifier "set0". The following example uses default "`notifications_direct`" routeConfig identifier configuration from UDR custom-values.yaml file. The notifications_direct, set0, notifications_direct_criteria_0, and notifications_direct_action_0 are custom identifiers used in the following example. If there are any change on the identifier name, you must modify all applicable strings in the custom-values.yaml.

```
egressgateway:
  routesConfig:
  # Route for Notification traffic
  - id: notifications_direct
    uri: http://dummy.dontchange
    path: /**
    order: 2
    metadata:
      httpsTargetOnly: false
      httpRuriOnly: false
      sbiRoutingEnabled: true
    filterName1:
      name: SbiRouting
      args:
        peerSetIdentifier: set0
        customPeerSelectorEnabled: false
        errorHandling:
         - errorCriteriaSet: notifications_direct_criteria_0
           actionSet: notifications_direct_action_0
           priority: 1
    filterNameControlShutdown:
      name: ControlledShutdownFilter
      args:
        applicableShutdownStates:
          - "COMPLETE_SHUTDOWN"
```

Configure sbiRoutingErrorCriteriaSets

Egress Gateway error criteria to validate non successful responses is configured in `sbiRoutingErrorCriteriaSets`. Only errors that match this criteria is eligible for reroute attempts. The following example uses `notifications_direct_criteria_0` is a custom

identifier. If there are any change on the identifier name, you must modify all applicable strings in the custom-values.yaml. By default UDR has the following `sbiRoutingErrorCriteriaSets`.

```
egressgateway:
  sbiRoutingErrorCriteriaSets:
   - id: notifications_direct_criteria_0
     method:
     - GET
     - POST
     - PUT
     - DELETE
     - PATCH
     exceptions:
     - java.util.concurrent.TimeoutException
     - java.net.UnknownHostException
```

Configure sbiRoutingErrorActionSets

Set the `attempts` as 2 in sbiRoutingErrorActionSets identifier that correspond to the `routesConfig.actionSet` identifier configured in section **Configure routesConfig**.

The following example uses default "`notifications_direct_action_0`" in sbiRoutingErrorActionSets identifier configuration from UDR custom values yaml file. Since the "set0" is configured with default "notifications_direct_action_0" identifier in section **Configure routesConfig**, you must set the `attempts` 2 to "notifications_direct_action_0" identifier. The notifications_direct_action_0 is the custom identifier used in the following example. If there are any change on the identifier name, you must modify all applicable strings in the custom-values.yaml.

```
egressgateway:
  sbiRoutingErrorActionSets:
   - id: notifications_direct_action_0
     action: reroute
     attempts: 2
     blackList:
      enabled: false
      duration: 60000
```

2.2.1.21 Configuring HPA and PDB in UDR Custom Value File

Perform the following steps to configure the supported HPA and PDB versions available on the CNE or any other container based environment where UDR is deployed:

1. Run the following command to check the supported Kubernetes resource versions.

   ```
   kubectl api-versions
   ```

2. Add the API versions to the custom values configurations as follows. The default API versions supported for HPA is `autoscaling/v2` and PDB is `policy/v1`.

```
k8sResources:
  hpat:
    supportedVersions:
    - autoscaling/v2
  pdb:
    supportedVersions:
    - policy/v1
```

2.2.1.22 Enabling and Disabling Conflict Resolution

To enable or disable the Conflict Resolution feature, see "Conflict Resolution" section in the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

2.2.2 Installation Tasks

This section explains how to install UDR.

> **Note**
>
> Before installing UDR, you must complete Prerequisites and PreInstallation Tasks.

2.2.2.1 Installing UDR Package

To install the UDR package, perform the following steps:

You can install UDR either with HELM repository or with HELM tar. To install UDR in Kubernetes cluster:

1. Run the following command to access the extracted package:

   ```
   cd ocudr-<release_number>
   ```

2. (Optional) Customize the `ocudr-custom-values-24.3.0.yaml` file with the required input parameters. To customize the file, see UDR Configuration Parameters.
3. Run the following command to install UDR:

   ```
   helm install <helm chart> [--version <OCUDR version>] --name <release> --
   namespace <k8s namespace> -f <ocudr-custom-values-24.3.0.yaml>
   ```

   In the above command:

   - *<helm chart>* - is the name of the chart, which is of the form <helm repo>/ocudr.
   - *<OCUDR version>* - is the software version (helm chart version) of the OCUDR. This is optional. If omitted, the default is latest version available in helm repository.
   - *<release>* - is a user defined release name to identify the helm deployment, all pod names, service name, deployment name will be prepended by this release name. This name should be different for each UDR/SLF site incase of multisite deployments
   - *<k8s namespace>* - is a name of user's choice to identify the Kubernetes namespace of the Unified Data Repository. All the Unified Data Repository microservices are deployed in this Kubernetes namespace.

- <ocudr-custom-values-24.3.0.yaml> - is the customized ocudr-custom-values-24.3.0.yaml file. The ocudr-custom-values-24.3.0.yaml file is a part of customer documentation. Download the file and modify it as per user site.

> **Note**
>
> If helm3 is used, run the following command to install UDR:
>
> ```
> helm install -name <release> --namespace <k8s namespace> -f <ocudr-custom-values-24.3.0.yaml> <helm chart> [--version <OCUDR version>]
> ```

> **Caution**
>
> Do not exit from helm install command manually. After running the helm install command, it takes some time to install all the services. In the meantime, you must not press "ctrl+c" to come out from helm install command. It leads to some anomalous behavior.

2.2.3 Post Installation Task

This section explains the postinstallation tasks for UDR.

2.2.3.1 Verifying Installation

To verify the installation:

1. Run the following command:

   ```
   helm ls release_name -n <release-namespace>
   ```

   Where,

   `<hem-release>` is the Helm release name of UDR.

   `<namespace>` is the namespace of UDR deployment.

   In the output, if `STATUS` is showing as `deployed`, then the deployment is successful.

2. Run the following command to verify if the pods are up and active:

   ```
   kubectl get jobs,pods -n release_namespace
   ```

   Where,

   `<Namespace>` is the namespace where UDR is deployed.

   ```
   kubectl get pod -n ocudr
   ```

   In the output, the `STATUS` column of all the pods must be `Running` and the `READY` column of all the pods must be `n/n`, where n is the number of containers in the pod.

3. Run the following command to verify if the services are deployed and active:

   ```
   kubectl get services -n release_namespace
   ```

   Example: `kubectl get services -n ocudr`

If the installation is unsuccessful or the `STATUS` of all the pods is not in the as `Running` state, perform the troubleshooting steps provided in the *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.

2.2.3.2 Performing Helm Test

This section describes how to perform sanity check for UDR installation through Helm test. The pods to be checked should be based on the namespace and label selector configured for the Helm test configurations.

> **Note**
>
> Helm test can be performed only on Helm3.

Helm Test - Overview

Helm Test is a feature that validates successful installation of UDR along with the readiness of all its pods. It also validates PVCs and confirms if they are in the bound state.

This test also checks for all the PVCs to be in the Bound state under the release namespace and label selector configured.

Configuring Helm Test

Before running the Hem Test command, it is important to do the following configurations:

1. Complete the Helm test configurations under the Global section of the `custom-values.yaml` file.

```
global:
  # Helm test related configurations
  test:
    nfName: ocudr
    image:
      name: ocudr/nf_test
      tag: 24.3.0
    config:
      logLevel: WARN
      timeout: 240
    resources:
    - horizontalpodautoscalers/v1
    - deployments/v1
    - configmaps/v1
    - prometheusrules/v1
    - serviceaccounts/v1
    - poddisruptionbudgets/v1
    - roles/v1
    - statefulsets/v1
    - persistentvolumeclaims/v1
```

```
- services/v1
- rolebindings/v1
#used to enable the helm test logging feature
complianceEnable: true
```

For more details about the configurable parameters, see UDR Configuration Parameters.

2. Ensure the `app.kubernetes.io/instance: {{ .Release.Name }}` label is part of all microservice deployments. The Helm Test feature takes the labelSelector internally, along with the helm release namespace, to select the pods and PVCs for verification.

> **Note**
>
> If the above label is not present, add it to the labels section so that the helm test can work on specific helm release.

3. Run the following helm test command:

```
helm test <helm_release_name> -n <k8s namespace>
```

Once the helm test job completes, check the output whether the test job is successful or not.

> **Note**
>
> - The Pod or PVC to be verified is fetched based on namespace and labelselector. If the list is empty, then the Helm Test is considered as success. If the Helm Test fails with errors, then see the *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
> - When the deployment has two replicas enabled for the nrf-client-nfmanagement micro service, the helm test fails stating that one of the nrf-client-nfmanagement pod is not ready. In such a case, this behavior is expected and the helm test failure can be ignored.

2.2.3.3 Taking a Backup

Take a backup of the following files, which are required during fault recovery:

- Updated `custom-values.yaml` file
- Updated Helm charts
- Secrets, certificates, and keys that are used during installation

3
Customizing UDR

This chapter provides information about customizing Oracle Communications Cloud Native Core, Unified Data Repository (UDR) deployment in a cloud native environment.

A sample `custom-values.yaml file` is available in Custom_Templates file. For more information on how to download the package, see Downloading Unified Data Repository Package.

3.1 UDR Configuration Parameters

This section includes information about the configuration parameters of UDR.

UDR allows customization of parameters for the different microservices and related settings.

> **Note**
>
> Mandatory parameters must be configured before the UDR deployment.

3.1.1 Global Configurable Parameters

The Global Configurable Parameters are suffixed to all the container names of UDR. These values are useful to add custom annotation(s) to all non-load balancer type services that UDR Helm chart creates.

The following table provides global configurable parameters:

Table 3-1 Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
dockerRegistry	Docker registry from where the images are pulled	ocudr-registry.us.oracle.com/ocudr	NA	NA
mysql.dbServiceName	Name of the database service	mysql-connectivity-service.occne-infra	NA	Its the connectivity service exposed by cnDBTier deployment.
mysql.port	Port exposed by database service	3306	NA	NA
mysql.configdbname	Database name for configdb	udrconfigdb	NA	NA
mysql.dbname	Database name for udrDB	udrdb	NA	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
mysql.dbUNameLiteral	Database user name for Literal	dsusername **Note**: This value must not be modified	NA	NA
mysql.dbPwdLiteral	Database password name for Literal	dspassword **Note**: This value must not be modified	NA	NA
mysql.dbEngine	Database engine name	NDBCLUSTER **Note**: This value must not be modified	NA	NA
nrfClientDbName	Database name for nrf-client service	configdbname **Note:** This value must not be modified	NA	
udrTracing.enable	Flag to enable udr tracing on Jaeger	false	true/false	NA
udrTracing.host	Jaegar service name installed on the cluster	occne-tracer-jaeger-agent.occne-infra	NA	NA
udrTracing.port	Jaegar service port installed on the cluster	14268	NA	NA
dbenc.shavalue	Encryption Key size	256	256 or 512	NA
serviceAccountName	Name of the service account	null	NA	The service account, role and role bindings required for deployment should be created prior to UDR installation. For more information see, creating service account.
configServerEnable	Flag to enable config-server	&configServerEnabled true	true/false	NA
dbCredSecretName	Name of the secret created for storing database credentials	ocudr-secrets	NA	NA
configServerFullNameOverride	Name of the Config-server to be used in deployment	nudr-config-server	NA	NA
configServicePort	HTTP port for nudr-config microservice	&configServiceHttpPort 5001	Valid Port	Keep the reference unchanged
createNetworkPolicy	Flag to enable Network Policy	false	true/false	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
alternatRouteServiceEnable	Flag to enable alternate route service	false	true/false	NA
appinfoServiceEnable	Flag to enable appinfo service.	true	true/false	NA
vendor	Logging parameter picked up by Kibana	Oracle	NA	NA
app_name	Logging parameter picked up by Kibana	alternate-route	NA	NA
performanceServiceEnable	Flag to enable performance info service	false	true/false	NA
servicePorts.perfInfoHttp	HTTP port exposed on performance info service	5905	valid port	NA
envJaegerQueryUrl	Jaeger URL to be used for perf-info service	http://occne-tracer-jaeger-query.occne-infra/	valid URL	NA
overloadManagerEnabled	Enabled when overload manager is required. By setting this flag to true, readiness check microservice provisions the overload configurations to database.	false	true/false	NA
sbiCorrelationInfoEnable	If the flag is enabled, then the 3gpp-Sbi-Correlation-Info header is added in the dr-service responses and notification requests in Egress Gateway.	false	true/false	NA
ephemeralStorage.requests.containersLogStorage	Log storage for all the containers used for ephemeral storage allocation request.	&containersLogStorageRequestsRef 50 Unit: MB	NA	Do not remove reference
ephemeralStorage.requests.containersCrictlStorage	Crictl storage for all the containers used for ephemeral storage allocation request.	&containersCrictlStorageRequestsRef 2 Unit: MB	NA	Do not remove reference

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
ephemeralStorage.limits.containersLogStorage	Log storage for all the containers used for ephemeral storage allocation.	&containersLogStorageLimitsRef 1000 Unit: MB	NA	Do not remove reference
ephemeralStorage.limits.containersCrictlStorage	Crictl storage for all the containers used for ephemeral storage allocation limits.	&containersCrictlStorageLimitsRef 2 Unit: MB	NA	Do not remove reference
udrServices	Services supported on UDR deployment. This configuration decides the schema execution on UDR database that is done by nudr-preinstall hook pod	All	All/nudr-dr/nudr-group-id-map/n5g-eir-eic	For SLF, set udrServices value as nudr-group-id-map. For EIR, set udrServices value as n5g-eir-eic.
consumerNF	Network functions that use the UDR. It is a comma-seperated list of Network functions.	If **udrServices** parameter is set to `All`; default value is PCF,UDM,NEF If **udrServices** parameter is set to `nudr-dr`; default value is PCF	PCF/UDM/NEF (Any combination of these 3)	
udrorAll	Services that need to be monitored by app info in "All" or "dr" mode	&allordr	nudr-drservice,nudr-dr-provservice,nudr-notify-service,nudr-diameterproxy,egressgateway,ingressgateway,nudr-config,nudr-config-server,alternate-route,nudr-ondemand-migration	NA
slf	Services that need to be monitored by app info in "Anudr-group-id-map" mode	&slf	nudr-drservice,nudr-dr-provservice,egressgateway,ingressgateway,nudr-config,nudr-config-server,alternate-route	NA
udsfEnable	Flag to enable UDSF services	true	true/false	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nfInstanceId	NF Instance ID of UDR (same is registered with NRF)	5a7bd676-ceeb-44bb-95e0-f6a55a328b03	Valid UUID	The reference definition remains unchanged. A valid Universally Unique Identifier (UUID) is a 128-bit unique number that helps NFs to identify NRF.
autoCreateSubscriber	Flag to enable or disable the auto creation of the subscriber when the PUT operation is performed on a new UEID (Unique Entity Identifier).	true	true/false	NA
addDefaultBillingDay				
diamGatewayEnable	Flag to enable or disable diamGateway service.	true	true/false	NA
diamGatewayRealm	diameter gateway service realm	oracle.com	valid realm	NA
diamGatewayIdentity	diameter gateway service identity	nudr.oracle.com	valid identity	NA
FourGPolicyConfiguration.vsaLevel	The data level where VSA (Vendor Specific Attribute) data holding the 4G policy content is added.	smpolicy	smpolicy/nssai/dnn	NA
FourGPolicyConfiguration.vsaDefaultBillingDay	The Billing day value on 4G policy data	1	NA	NA
FourGPolicyConfiguration.snssai	The snssai value configurable on 4G policy data	2-FFFFFF	NA	NA
FourGPolicyConfiguration.dnn	The dnn value configurable on 4G policy data	dnn1	NA	NA
FourGPolicyConfiguration.ondemandMigration.enabled	Flag to enable on-demand migration service	&onDemandMigrationEnabled false	true/false	Keep the reference variable unchanged
FourGPolicyConfiguration.ondemandMigration.diamRealm	ondemand migration tool realm	oracle.com	valid realm	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
FourGPolicyConfiguration.ondemandMigration.diamIdentity	ondemand migration tool identity	ondemand-migration.nudr.oracle.com	valid identity or FQDN	Use migration keyword in the diameter identity
FourGPolicyConfiguration.ondemandMigration.deleteSubscriberOnSource	Enable or disable 4G Subscriber deletion	false	true or false	NA
FourGPolicyConfiguration.ondemandMigration.restEndpoint	4G Rest endpoint for deletion of subscriber	http://source-end-rest-endpoint	Valid Endpoint	NA
FourGPolicyConfiguration.sourceUdr.realm	Source diameter server realm	tekelec1.com	valid diameter realm	NA
FourGPolicyConfiguration.sourceUdr.identity	Source diameter server identity	local1.tekelec1.com	valid diameter identity	NA
FourGPolicyConfiguration.sourceUdr.hostIp	Source IP, refers to 4G UDR		valid HostIP or FQDN	NA
FourGPolicyConfiguration.sourceUdr.hostPort	Source Port, refers to 4G UDR	3868	Valid diameter port	NA
serviceMeshCheck	Flag to check side car container readiness when deployed with Aspen Service Mesh. It is enabled when deployed with Aspen Service Mesh.	&serviceMeshFlag false	true/false	Keep the reference variable unchanged
istioSidecarQuitUrl	Quit URL configurable for side car	&istioQuitUrl "http://127.0.0.1:15020/quitquitquit"	Valid URL	Keep the reference variable unchanged
istioSidecarReadyUrl	Readiness URL configurable for side car	&istioReadyUrl "http://127.0.0.1:15000/ready"	Valid URL	The port used should be the admin configured for istio proxy container. Keep the reference variable unchanged

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
test.nfName	NF name on which the helm test is performed. For UDR, the default value is UDR and is used in container name as suffix	UDR	NA	NA
test.image.name	Image name for helm test container image	nf_test	NA	NA
test.image.tag	Image version tag for helm test	24.3.2	NA	NA
test.config.logLevel	Log level for helm test pod	WARN	Possible Values - WARN INFO DEBUG	NA
test.config.timeout	Timeout value for helm test operation. If the timeout value exceeds, helm test is considered as failure	240	Range: 1-300 Unit:seconds	NA
test.resources	Kubernetes resources for which the API version information needs to be fetched	horizontalpodautoscalers/v1 • deployments/v1 • configmaps/v1 • prometheusrules/v1 • serviceaccounts/v1 • poddisruptionbudgets/v1 • roles/v1 • statefulsets/v1 • persistentvolumeclaims/v1 • services/v1 • rolebindings/v1	NA	NA
test.complianceEnable	Indicates whether or not the Kubernetes logging feature is enabled	true	true or false	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
preinstall.image.name	Image name for the nudr-prehook pod which takes care of database and table creation for UDR deployment	nudr_common_hooks	NA	NA
preinstall.image.tag	Image version for nudr-pre-install-hook pod image	24.3.0	NA	NA
preinstall.config.logLevel	Log level for pre-install hook pod	WARN	Possible Values - WARN INFO DEBUG	NA
preinstall.createUser	Flag to enable user creation on SQL nodes as part of pre-install hook.	false **Note:** Do not change the default value.	true/false	NA
preinstall.serviceMeshCheck	Flag to check side car container readiness when deployed with Aspen Service Mesh. Set this parameter to false if side car is not enabled only for this hook.	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference
preinstall.istioSidecarQuitUrl	Quit URL configurable for side car. Change its value only if the URL for this hook is different from the one configured in global section	*istioQuitUrl	NA	Do not change this value unless it is different from the value configured in global section reference.
preinstall.istioSidecarReadyUrl	Readiness URL configurable for side car. Change its value only if the URL for this hook is different from the one configured in global section	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
userCreation.image.name	Image name for the nudr-usercreation-hook pod for creating the database and table for UDR deployment.	nudr_common_hooks	NA	NA
userCreation.image.tag	Image version for nudr-usercreation-hook pod image.	24.3.0	NA	NA
userCreation.createUser	Creates user and grants UDR database access	true **Note:** Do not change this value.	true or false	NA
userCreation.config.logLevel	Log level for nudr-usercreation-hook pod.	WARN	Possible Values: WARN INFO DEBUG	
userCreation.serviceMeshCheck	Flag to check the sidecar container readiness when deployed with aspen service mesh. For more information, see serviceMeshCheck under global section in ocudr_custom_values file. **Note:** You must change the userCreation.serviceMeshCheck flag to false if sidecar is disabled, do this only if it is different from the value configured in the global section.	*serviceMeshFlag	NA	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
userCreation.istioSidecarQuitUrl	It is the quit URL configurable for sidecar. For more information, see istioSidecarQuitUrl under global section in ocudr_custom_values file. **Note:** You must change the URL of this hook only if it is different from the value configured in the global section.	*istioQuitUrl	NA	NA
userCreation.istioSidecarReadyUrl	It is the readiness URL configurable for sidecar. For more information, see istioSidecarReadyUrl under global section in ocudr_custom_values file. **Note:** You must change the URL of this hook only if it is different from the value configured in the global section.	*istioReadyUrl	NA	
preUpgrade.image.name	Image name of nudr-pre-upgrade pod that takes care of database schema updates on UDR datastore	nudr_common_hooks	NA	NA
preUpgrade.image.tag	Image version of nudr-pre-upgrade pod image	24.3.0	NA	NA
preUpgrade.config.logLevel	Log level of nudr-pre-upgrade hook pod	WARN	Possible Values: WARN INFO DEBUG	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
preUpgrade.serviceMeshCheck	Flag to check side car container readiness when deployed with Aspen Service Mesh. Set this parameter to false if side car is not enabled only for this hook.	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference
preUpgrade.istioSidecarQuitUrl	Quit URL configurable for side car. Update this parameter only if the URL of this hook is different from the one configured in global section	*istioQuitUrl	NA	Do not change this value unless it is different from the value configured in global section reference
preUpgrade.istioSidecarReadyUrl	Readiness URL configurable for side car. Update this parameter only if the URL of this hook is different from the one configured in global section	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference
preUpgrade.upgradeFailed	Flag to enable delete configuration database.	false	true/false	Change to true if previous upgrade failed and try to upgrade again.
postInstall.image.name	Image name for the nudr-post-upgrade pod, which will take care of schema updates on the UDR datastore	nudr_common_hooks	NA	NA
postInstall.image.tag	Image version for nudr-post-upgrade pod image	24.3.0	NA	NA
postInstall.config.logLevel	Log level for nudr-post-upgrade hook pod	WARN	Possible Values: WARN INFO DEBUG	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
postInstall.serviceMeshCheck	Flag to check the side car container readiness when deployed with Aspen Service mesh. For more information, refer to serviceMeshCheck under global section.	*serviceMeshFlag	NA **Note:** Do not change this value unless it is different from the value configured in global section reference.	NA
postInstall.istioSidecarQuitUrl	Quit URL configurable for sidecar. For more information, refer to istioSidecarQuitUrl under global section. Change only if the URL for this hook is different from the one configured in global section.	*istioQuitUrl	NA **Note:** Do not change this value unless it is different from the value configured in global section reference.	NA
postInstall.istioSidecarReadyUrl	Readiness URL configurable for sidecar. For more information, refer to istioSidecarReadyUrl under global section. Change only if the URL for this hook is different from the one configured in global section.	*istioReadyUrl	NA **Note:** Do not change this value unless it is different from the value configured in global section reference.	NA
postInstall.serviceToCheck	Services to check before adding default configuration for overload control	nudr-config-server,nudr-config	NA	NA
postInstall.serviceCheckTimeout	Service check timeout for servicesToCheck configuration above	100	NA	
postUpgrade.image.name	Image name for the nudr-post-upgrade pod, which takes care of schema updates on the UDR datastore.	nudr_common_hooks	NA	NA
postUpgrade.image.tag	Image version for nudr-post-upgrade pod image	24.3.0	NA	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
postUpgrade.config.logLevel	Log level for nudr-post-upgrade hook pod	WARN	Possible Values: WARN INFO DEBUG	NA
postUpgrade.serviceMeshCheck	Flag to check side car container readiness when deployed with Aspen Service mesh. Change to false if side car is disabled only for this hook	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in the global section reference.
postUpgrade.istioSidecarQuitUrl	Quit URL configurable for side car. Change only if the URL for this hook is different from the one configured in global section.	*istioQuitUrl	NA	Do not change this value unless it is different from the value configured in the global section reference.
postUpgrade.istioSidecarReadyUrl	Readiness URL configurable for side car. Change only if the URL for this hook is different from the one configured in global section.	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in the global section reference.
postRollback.image.name	Image name for the nudr-post-upgrade pod, which takes care of schema updates on the UDR datastore.	nudr_common_hooks	NA	NA
postRollback.image.tag	Image version for nudr-post-upgrade pod image	24.3.0	NA	NA
postRollback.config.logLevel	Log level for nudr-post-upgrade hook pod	WARN	Possible Values - WARN INFO DEBUG	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
postRollback.serviceMeshCheck	Flag to check side car container readiness when deployed with Aspen Servicemesh. Referred to serviceMeshCheck under global section. Change to false if side car is disabled only for this hook	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference
postRollback.istioSidecarQuitUrl	Quit url configurable for side car. Referred to istioSidecarQuitUrl under global section. Change only if the url for this hook is different from the one configured in global section	*istioQuitUrl	NA	Do not change this value unless it is different from the value configured in global section reference
postRollback.istioSidecarReadyUrl	Readiness url configurable for side car. Referred to istioSidecarReadyUrl under global section. Change only if the url for this hook is different from the one configured in global section	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference
nrfClientNfDiscoveryEnable	Flag to enable NRF client discovery pod **Note:** Do not enable the `nrfClientNfDiscoveryEnable` flag. This must be kept disabled. Discovery service is not used for UDR/SLF/EIR deployments.	false	true/false	NA
nrfClientNfManagementEnable	Flag to enable NRF client management pod	true	true/false	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
deploymentNrfClientService.envNfNamespace	Namespace of the UDR deployment	myudr	NA	NA
deploymentNrfClientService.envNfType	Name of nfType	udr	NA	NA
deploymentNrfClientService.envConsumeSvcName	The services to be monitored by perf-info to report load and capacity of configured NFServices. The format shall be <Service Name of the NF : the NFService name as per 29.510>	'nudr-drservice:nudr-drservice'	NA	NA
initContainerEnable	Flag to enable init container. The value of this parameter should be set to false when UDR is deployed with Aspen Service Mesh.	true	true/false	NA
nfName	Set to prefix of the name of microservices	nudr	NA	NA
applicationName	Application Name	ocudr	NA	NA
egressGatewayFullnameOverride	Egress gateway Host. It should be used as {{ ReleaseName }}-egressGatewayFullnameOverride	egressgateway	NA	NA
egressGatewayPort	Egress gateway Port	8080	Valid Port	NA
hookJobResources.limits.cpu	CPU limit for pods created in Kubernetes hooks or jobs created as part of UDR installation	1	NA	Applicable to helm test job as well

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
hookJobResources .limits.memory	Memory limit for pods created in Kubernetes hooks or jobs created as part of UDR installation	1Gi	NA	Applicable for helm test job as well
hookJobResources .limits.ephemeral-storage	Ephemeral storage limits for hook pods	1024Mi Unit MB	NA	NA
hookJobResources .requests.cpu	CPU requests for pods created in Kubernetes hooks or jobs created as part of UDR installation	0.5	NA	CPU allocated for hooks during deployment. It is applicable for helm test job as well.
hookJobResources .requests.memory	Memory requests for pods created in Kubernetes hooks or jobs created as part of UDR installation	0.5Gi	NA	The memory to be allocated for hooks during deployment. It is applicable for helm test job as well.
hookJobResources .requests.ephemeral-storage	Ephemeral storage request for hook pods	72Mi Unit MB	NA	NA
tlsVersionDowngrade	Downgrade tls version to TLS 1.2 while running the microservices. Retain its default value unless you face any issue w.r.t SSL version on microservices	false	true/false	NA
customExtension.allResources.labels	Custom Labels added to all the UDR Kubernetes resources	null	NA	This can be used to add custom label(s) to all Kubernetes resources that UDR helm chart creates.

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
customExtension.allResources.annotations	Custom Annotations added to all the UDR Kubernetes resources	null	NA	This can be used to add custom annotation(s) to all Kubernetes resources that UDR helm chart creates. ASM related annotations needs to be added under ASM Specific Configuration section.
customExtension.lbServices.labels	Custom Labels added to UDR Services that are of Load Balancer type	null	NA	This can be used to add custom label(s) to all Load Balancer type services UDR helm chart creates.
customExtension.lbServices.annotations	Custom Annotations added to UDR Services that are of Load Balancer type	null	NA	This can be used to add custom annotation(s) to all Load Balancer type services that UDR helm chart creates.
customExtension.lbDeployments.labels	Custom Labels added to UDR Deployment that is associated to a service of Load Balancer type	null	NA	This can be used to add custom label(s) to all Deployments that UDR helm chart creates which are associated to a service which is of Load Balancer type.
customExtension.lbDeployments.annotations	Custom Annotations added to UDR Deployment that is associated to a Service of Load Balancer type	null	NA	ASM related annotations needs to be added under ASM Specific Configuration section. This can be used to add custom annotation(s) to all Deployments that UDR helm chart creates which are associated to a service which if of Load Balancer type.

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
customExtension.nonlbServices.labels	Custom Labels added to UDR Services that is not of Load Balancer type	null	NA	This can be used to add custom label(s) to all non-Load Balancer type services that UDR helm chart creates.
customExtension.nonlbServices.annotations	Custom Annotations added to UDR Services that is not of Load Balancer type	null	NA	This can be used to add custom annotation(s) to all non-Load Balancer type services that UDR helm chart creates.
customExtension.nonlbDeployments.labels	Custom Labels added to UDR Deployment that are associated to a Service which is not of Load Balancer type	null	NA	This can be used to add custom label(s) to all Deployments that are created by UDR helm chart which are associated to a service which if not of Load Balancer type.
customExtension.nonlbDeployments.annotations	Custom Annotations added to UDR Deployment that are associated to a Service which is not of Load Balancer type	null	NA	ASM related annotations to be added under ASM Specific Configuration section This can be used to add custom annotation(s) to all Deployments that are created by UDR helm chart which are associated to a service which if not of Load Balancer type.
k8sResource.container.prefix	Value that is prefixed to all the container names of UDR	null	NA	NA
k8sResource.container.suffix	Value that is suffixed to all the container names of UDR	null	NA	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainers	Flag to enable addition of container configuration under extraContainersTpl to all the deployments under UDR umbrella. This parameter is used for debug container template	DISABLED	Allowed values: • ENABLED • DISABLED	This configuration is dependent on extraContainers configuration under each microservice. This parameter is not used for hooks.

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainerTpl	The template for extraContainer is defined in this configuration. For now, it defines the debugContainer template. It is applicable to all microservices under UDR umbrella	`- command: - /bin/sleep - infinity image: {{ .Values.global.dockerRegistry }}/debug-tools:24.3.1 imagePullPolicy: IfNotPresent name: {{ printf "%s-tools-%s" (include "getprefix" .) (include "getsuffix" .) | trunc 63 | trimPrefix "-" | trimSuffix "-" }} resources: requests: ephemeral-storage: "512Mi" cpu: "0.5" memory: "4Gi" limits: ephemeral-storage: "512Mi"`	NA	This is not used for hooks.

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		cpu: "0.5" memory: "4Gi" securityContext: allowPrivilegeEscalation: true capabilities: drop: - ALL add: - NET_RAW - NET_ADMIN runAsUser: 1002 volumeMounts: - mountPath: /tmp/tools name: debug-tools-dir		

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

extraContainersVolumesTpl	Debug container volume template	`- name: debug-tools-dir`
		
		`emptyDir:`
		
		`medium: Memory`
		
		`sizeLimit: "4Gi"`		

tolerationsSetting	Flag to enable the toleration setting at the global level. If set to ENABLED and local toleration setting at each microservice level is set to USE_GLOBAL_VALUE the same can be used in the microservice.	DISABLED	Allowed Values: • ENABLED • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the global level. If set to ENABLED and local nodeSelection setting at each microservice level is set to USE_GLOBAL_VALUE the same can be used in the microservice.	DISABLED	Allowed Values: • ENABLED • DISABLED	NA

tolerations	When global.tolerationSetting is ENABLED configure toleration here.	[]	Example:
			
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelector.nodeKey	NodeSelector key configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion at microservice level is set to v1 then this configuration comes to use. And this configuration does not depend on nodeSelection flag, once configured this can be used for all microservices. If at microservice level this is configured with a different value, then the same can be used for that particular microservice.	' '	Valid key for a label for a particular node.	NA
nodeSelector.nodeValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion at microservice level is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this can be used for all microservices. If at microservice level this is configured with a different value, then the same can be used for that particular microservice	' '	Valid value pair for the above key for a label for a particular node.	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion at microservice level is set to v2. Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod.	NA

hooksCommonConfig.tolerations	When global.tolerationSetting is ENABLED configure tolerations here.	[]	Example:
			
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2	NA
hooksCommonConfig.nodeSelector.nodeKey	Configuring NodeSelector key for UDR hooks. This excludes the common service hooks when hooksCommonConfig is set to v1 then this configuration comes to use. `helmBasedConfigurationNodeSelectorApiVersion`	''	Valid key for a label for a particular node.	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
hooksCommonConfig.nodeSelector.nodeValue	NodeSelector value configuration for UDR hooks, This excludes the common service hooks when hooksCommonConfig is set to v1 then this configuration comes to use. `helmBasedConfigurationNodeSelectorApiVersion`	''	Valid value pair for the above key for a label for a particular node.	NA
hooksCommonConfig.nodeSelector	Configuring NodeSelector when hooksCommonConfig is set to v2. `helmBasedConfigurationNodeSelectorApiVersion at microservice level` Uncomment and use this configuration when required. Else, keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod.	NA
configClientConnectTimeout	It is the value in milliseconds that determines the maximum allowed time to connect to the configuration client API. Note: Do not change the default value unless required.	1000	Any positive integer	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
configClientReadTimeout	It is the value in milliseconds that determines the maximum allowed time read data from configurationg client API. Note: Do not change the default value unless required.	1000	Any positive integer	NA
subscriberActivityEnabled	Flag to enable or disable the subscriber activity logging feature	true or false	false	NA
autoEnrolOnSignalling	Flag to enable or disable auto enrolment of the subscriber when PUT operation is executed on UsageMonitoringInformation or PATCH operation on SM Data.	true	true or false	NA
etagEnabled	Flag to enable or disable Etag support for SM Data	false	true or false	NA
enableControlledShutdown	Flag to enable or disable operational state changes.	false	true or false	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
k8sResources.hpat .supportedVersions	Supported versions for horizontal pod autoscalers. This is updated as per the latest CNE version which has kubernetes version 1.25. For older CNE versions or different container based environments on Kubernetes or openshift containers, you mush check the supported versions for HPA resource and add to the default list.	autoscaling/v2	NA	NA
k8sResources.pdb. supportedVersions	Supported versions for pod disruption budget For older CNE versions or different container based environments on Kubernetes or openshift containers, you mush check the supported versions for PDB resource and add to the default list.	policy/v1	NA	NA
multipleIgwEnabled	Flag to include multiple criteria, such as TPS and signaling connection, in NF Scoring calculation	true	true or false	NA
quotaVersion	This value is used to configure the quota entity version in VSA data	3	NA	NA
DynamicQuotaVersion	This value is used to configure the dynamic quota entity version in VSA data	3	NA	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
dbConflictResolutionEnabled	The configuration enables conflict resolution for UDR subscriber database when installed on a multiple site setup.	false	true or false	NA
maxNumberOfSubscriptions	Limits the number of subscriptions for one subscriber key. Applicable to subs-to-notify and SNR requests.	30	NA	NA
suppressNotificationEnabled	Flag to enable or disable the Suppress Notification feature.	true	true or false	NA
nfType	Configure the NF type using Helm charts for error detail content.	UDR	NA	NA
nfFqdn	Configure the NF FQDN using Helm charts for error detail content.	ocudr-ingressgateway-sig.oracle.com	Valid FQDN	NA
s13InterfaceEnable	Configuration to enable or disable the Diameter S13 Interface feature. The flag is set to `false` by default. When set to true, Mobile Equipment Identity Check Request (ECR) and Equipment Identity Answer (ECA) requests is sent.	false	true or false	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
eirConfigurations.defaultResponse	Configuration to set the EIC (Equipment Identity Check) response. When there is a Permanent Equipment Identifier (PEI) match found on EIR but the query parameters Subscription Permanent Identifier (SUPI) or Generic Public Subscription Identifier (GPSI) do not match, the default value will be EQUIPMENT UNKNOWN response. This can be updated to WHITELISTED, GREYLISTED, or BLACKLISTED using this configuration.	EQUIPMENT_UNKNOWN	GREYLISTED, BLACKLISTED, WHITELISTED, or EQUIPMENT_UNKNOWN	NA
eirConfigurations.accessLogEnabled	Flag to enable or disable the Access Log feature	true	true or false	NA
eirConfigurations.defaultResponseIMEINotFound	Configuration to set the EIC response. When PEI is match not found on database, the default value will be EQUIPMENT UNKNOWN response. This can be updated to WHITELISTED, GREYLISTED, or BLACKLISTED using this configuration.	EQUIPMENT_UNKNOWN	GREYLISTED, BLACKLISTED, WHITELISTED, or EQUIPMENT_UNKNOWN	NA

Table 3-1 (Cont.) Global Configurable Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
eirConfigurations.imsiLookupFallbackEnabled	Configuration to enable or disable the IMSI Lookup Fallback feature. The default value is set as `false`. This feature can be configured using Helm, REST APIs, or CNC Console.	false	true or false	NA

3.1.2 nudr-drservice Microservice Parameters

Following table provides the parameters for nudr-drservice microservice.

Table 3-2 nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
image.name	Docker Image name	nudr_datarepository_service	NA	NA
image.tag	Tag or version of the Image	24.3.0	NA	NA
image.pullPolicy	Indicates whether image needs to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
logging.level.root	Log level of the nudr-drservice pod	WARN	Possible Values - WARN INFO DEBUG	NA
deployment.replicaCount	Number of nudr-drservice pods to be maintained by replica set created with deployment	2	NA	NA
minReplicas	Minimum number of pods	2	NA	NA
maxReplicas	Maximum number of pods	8	NA	NA
maxUnavailable	Number of replicas that can go down during a disruption	1	NA	This parameter is used for PodDisruption Budget k8s resource

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.type	Kubernetes service type for exposing UDR deployment	ClusterIP	Possbile Values- ClusterIP NodePort LoadBalancer	It is recommended to set the default value as ClusterIP always.
service.port.http	The http port to be used in nudr-drservice service	5001	NA	NA
service.port.https	The https port to be used for nudr-drservice service	5002	NA	NA
service.port.management	The actuator management port to be used for nudr-drservice service	9000	NA	NA
resources.requests.cpu	Number of CPUs allocated for nudr-drservice pod during deployment	2	NA	NA
resources.requests.memory	The memory to be allocated for nudr-drservice pod during deployment	2Gi	NA	NA
resources.requests.logStorage	Log storage for ephemeral storage allocation request	*containersLogStorageRequestsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.requests.crictlStorage	Crictl storage for ephemeral storage allocation request	*containersCrictlStorageRequestsRef Unit: MB	NA	Change the reference to a different value if allocation needs to be different for this container
resources.limits.cpu	CPU limit allotted	2	NA	NA
resources.limits.memory	Memory limit allotted	2Gi	NA	NA
resources.limits.logStorage	Log storage for ephemeral storage allocation limit	*containersLogStorageLimitsRef Unit: MB	NA	Change the reference to a different value if allocation needs to be different for this container

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.limits.crictlStorage	Crictl storage for ephemeral storage allocation limit	*containersCrictlStorageLimitsRef Unit: MB	NA	Change the reference to a different value if allocation needs to be different for this container
resources.target.averageCpuUtil	CPU utilization limit for autoscaling to create HPA	80	NA	NA
notify.port.http	HTTP port on which notify service is running	5001	NA	NA
notify.port.https	HTTPS port on which notify service is running	5002	NA	NA
serviceToCheck	Essential services on which init container performs readiness check	nudr-config-server,nudr-config	Comma separated microservice names	NA
hikari.poolsize	Mysql Connection pool size The hikari pool connection size to be created at start up	100	NA	NA
hikari.connectionTimeout	MYSQL connection timeout	1000 Unit: Milliseconds	NA	NA
hikari.minimumIdleConnections	Minimum idle MYSQL connections maintained	2	Valid. No, less than poolsize configured	NA
hikari.idleTimeout	Idle MYSQL connections timeout	10000 Unit: Milliseconds	NA	NA
hikari.queryTimeout	Query Timeout for a single database query. If set to -1, there will be no timeout set for database queries.	-1 Unit: Seconds	NA	NA
internalTracingEnabled	Flag to enable Jaeger tracing for nudr-drservice pod	false	true/false	NA
maxConcurrentPushedStreams	Maximum number of concurrent requests that can be pushed per destination	6000	NA	NA

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
maxRequestsQueuedPerDestination	Maximum number of requests that can be queued per destination	5000	NA	NA
maxConnectionsPerDestination	Maximum connection allowed per destination	10	NA	NA
maxConnectionsPerIp	Maximum connection allowed per IP	10	NA	NA
requestTimeout	Client request timeout	5000 Unit: Milliseconds	NA	NA
connectionTimeout	Client connection timeout	10000 Unit: Milliseconds	NA	NA
idleTimeout	Client connection idle timeout	0 Unit: Milliseconds	NA	NA
dnsRefreshDelay	Server DNS refresh delay	120000 Unit: Milliseconds	NA	NA
pingDelay	Server ping delay	60 Unit: Milliseconds	NA	NA
connectionFailureThreshold	Client connection failure threshold	10	NA	NA
jettyserviceMeshCheck	Flag to enable or disable jetty service mesh check for ASM deployment	false	true or false	NA
service.customExtension.labels	This can be used to add custom label(s) to nudr-drservice Service	null	NA	NA
service.customExtension.annotations	This can be used to add custom annotation(s) to nudr-drservice Service	null	NA	NA
deployment.customExtension.labels	This can be used to add custom label(s) to nudr-drservice Deployment	null	NA	NA

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	This can be used to add custom annotation(s) to nudr-drservice deployment	`sidecar.istio.io/proxyCPU: "2000m"`
		`sidecar.istio.io/proxyCPULimit: "2000m"`
		`sidecar.istio.io/proxyMemory: "1Gi"`
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		`proxy.istio.io/config: |`
		
		`terminationDrainDuration: 60s`
		
		`concurrency: 8`
			NA	NA

startupProbe.failureThreshold	It is the configurable number of times the startup probe is retried for failures. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	40 Unit: Seconds	NA	NA
startupProbe.periodSeconds	It is the time interval for every startup probe check. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	5 Unit: Seconds	NA	NA

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainers	This configuration decides service level extraContainer support. The default value is dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA
serviceMeshCheck	Flag to check service mesh. Set to false when side car is not included for this service	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section
istioSidecarReadyUrl	Readiness url configurable for side car. Change this url only if it is different from the side car container url in this microservice	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section
defaultGroupIdEnabled	Flag to enable or disable the Default Group ID feature	False	True or False	NA
defaultSLFGroupName	The default group name provisioned. If there are any changes done to this through CNC console, then the default group should be reprovisioned with the modified group name. But if the change is done through helm values and helm upgrade is performed, then the default group will be created with the updated name.	DefaultGrp	True or False	NA

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level . If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

tolerations	When tolerationsSetting is ENABLED configure tolerations here.	[]	NA
			Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2	NA
nodeSelector.nodeKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid key for a label for a particular node	NA
nodeSelector.nodeValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid value pair for the above key for a label for a particular node	NA
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2 Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod	NA
subscriptionDataSubscriptionsOnly	NA	False	NA	NA
onDemandMigrationEnabled	Flag to enable or disable the on demand migration service.	False	True or False	NA

Table 3-2 (Cont.) nudr-drservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
convertUeidToStandard	Enable to convert ueId key to standard format while provisioning or updating profile.	false	True or False	NA
onDemandMigrationSigEnabled	Flag to enable or disable on-demand migration for provisioning interface	false	true or false	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	60s Unit: Seconds	NA	NA
notificationThreadPool.corePoolSize	Thread pool configurations for notification processing. **Note**: This is engineering configurations, you must not change the default value.	100	NA	NA
notificationThreadPool.maxPoolSize	Thread pool configurations for notification processing. **Note**: This is engineering configurations, you must not change the default value.	200	NA	NA
notificationThreadPool.maxQueueCapacity	Thread pool configurations for notification processing. **Note**: This is engineering configurations, you must not change the default value.	5000	NA	NA

3.1.3 nudr-dr-provservice Microservice Parameters

Following table provides the parameters for nudr-dr-provservice microservice.

Table 3-3 nudr-dr-provservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
image.name	Docker Image name	nudr_datarepository_service	NA	NA
image.tag	Tag of Image	24.3.0	NA	NA
image.pullPolicy	This setting allows you to determine whether image need to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
logging.level.root	Log level of the nudr-drservice pod	WARN	Possible Values - WARN INFO DEBUG	NA
deployment.replicaCount	Number of nudr-drservice pods to be maintained by replica set created with deployment	2	NA	NA
minReplicas	Minimum number of pods	2	NA	NA
maxReplicas	Maximum number of pods	8	NA	NA
maxUnavailable	Number of replicas that can go down during a disruption	1	NA	This parameter is used for PodDisruption Budget Kubernetes resource
service.type	The Kubernetes service type for exposing UDR deployment	ClusterIP	Possible Values- • ClusterIP • NodePort • LoadBalancer	Suggested to be set as ClusterIP (default value) always
service.port.http	The HTTP port to be used in nudr-drservice service	5001	NA	NA
service.port.https	The HTTPS port to be used for nudr-drservice service	5002	NA	NA
service.port.management	The actuator management port to be used for nudr-drservice service	9000	NA	NA
resources.requests.cpu	The CPU to be allocated for nudr-drservice pod during deployment	2	NA	NA
resources.requests.memory	The memory to be allocated for nudr-drservice pod during deployment	2Gi	NA	NA

Table 3-3 (Cont.) nudr-dr-provservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.requests.logStorage	Log storage for ephemeral storage allocation request	*containersLogStorageRequestsRef Unit: MB	NA	Change the reference to a different value if allocation needs to be different for this container
resources.requests.crictlStorage	Crictl storage for ephemeral storage allocation request	*containersCrictlStorageRequestsRef Unit: MB	NA	Change the reference to a different value if allocation needs to be different for this container
resources.limits.cpu	Cpu allotment limitation	2	NA	NA
resources.limits.memory	Memory allotment limitation	2Gi	NA	NA
resources.limits.logStorage	Log storage for ephemeral storage allocation limit	*containersLogStorageLimitsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.limits.crictlStorage	Crictl storage for ephemeral storage allocation limit	*containersCrictlStorageLimitsRef Unit: MB	NA	Change the reference to a different value if allocation needs to be different for this container
resources.target.averageCpuUtil	CPU utilization limit for creating HPA	80	NA	NA
notify.port.http	HTTP port on which notify service is running	5001	NA	NA
notify.port.https	HTTPS port on which notify service is running	5002	NA	NA
serviceToCheck	Essential services on which init container will perform readiness checks.	nudr-config-server,nudr-config	Comma separated microservice names	NA
hikari.poolsize	The hikari pool connection size to be created at start up	100	NA	NA
hikari.connectionTimeout	MYSQL connection timeout	1000 Unit: Milliseconds	Unit: Milliseconds	NA
hikari.minimumIdleConnections	Minimum idle MYSQL connections maintained	2	Valid No, less than poolsize configured	NA

Table 3-3 (Cont.) nudr-dr-provservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
hikari.idleTimeout	Idle MYSQL connections timeout	10000 Unit: Milliseconds	NA	NA
hikari.queryTimeout	Query timeout for a single databse query. If set to -1, there will be no timeout set for database queries.	-1 Unit: Seconds	NA	NA
internalTracingEnabled	Flag to enable/disable jaeger tracing for nudr-drservice	false	true/false	NA
maxConcurrentPushedStreams	Maximum number of concurrent requests that can be pushed per destination	6000	NA	NA
maxRequestsQueuedPerDestination	Maximum number of requests that can be queued per destination	5000	NA	NA
maxConnectionsPerDestination	Maximum connection allowed per destination	10	NA	NA
maxConnectionsPerIp	Maximum connection allowed per IP	10	NA	NA
requestTimeout	Client request timeout	5000 Unit: Milliseconds	NA	NA
connectionTimeout	Client connection timeout	10000 Unit: Milliseconds	NA	NA
idleTimeout	Client connection idle timeout	0 Unit: Milliseconds	NA	NA
dnsRefreshDelay	Server DNS refresh delay	120000 Unit: Milliseconds	NA	NA
pingDelay	Server ping delay	60 Unit: Milliseconds	NA	NA
connectionFailureThreshold	Client connection failure threshold	10	NA	NA
jettyserviceMeshCheck	Flag to enable or disable jetty service mesh check for ASM deployment	false	true/false	NA
service.customExtension.labels	This can be used to add custom label(s) to nudr-drservice Service.	null	NA	NA
service.customExtension.annotations	This can be used to add custom annotation(s) to nudr-drservice Service.	null	NA	NA
deployment.customExtension.labels	This can be used to add custom label(s) to nudr-drservice Deployment.	null	NA	NA

Table 3-3 (Cont.) nudr-dr-provservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	This can be used to add custom annotation(s) to nudr-drservice deployment.	`sidecar.istio.io/proxyCPU: "2000m"`
		`sidecar.istio.io/proxyCPULimit: "2000m"`
		`sidecar.istio.io/proxyMemory: "1Gi"`
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		`proxy.istio.io/config: |`
		`terminationDrainDuration: 60s`
		`concurrency: 8`
			NA	NA

startupProbe.failureThreshold	It is the configurable number of times the startup probe is retried for failures. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	40 Unit: Seconds	NA	NA
startupProbe.periodSeconds	It is the time interval for every startup probe check. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	5 Unit: Seconds	NA	NA
extraContainers	This configuration decides at this microservice level the extra container need to be used or not. The default value lets it to be dependent on what is configured at the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA

Table 3-3 (Cont.) nudr-dr-provservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
serviceMeshCheck	Enable when deployed in serviceMesh, refer to the serviceMeshCheck in global section. It is set as false when side car is not included for this service	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference
istioSidecarReadyUrl	Readiness URL configurable for side car, refer to global section istioSidecarReadyUrl. Change this value only if the URL is different for the side car container in this microservice	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference
onDemandMigrationEnabled	Flag to enable or disable the on demand migration service	False	True or False	NA
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level . If set to USE_GLOBAL_VALUE, the global.tolerations configured value is used. If set to ENABLED, the tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the the microservice level or global level . If set to USE_GLOBAL_VALUE, the global.tolerations configured value is used. If set to ENABLED, the tolerations configuration below in the same section is used. This is applicable for the v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

Table 3-3 (Cont.) nudr-dr-provservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

tolerations	When tolerationsSetting is ENABLED, tolerations are configured here.	[]	NA
			Example:
			
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

helmBasedConfigurationNodeSelectorApiVersion	Indicates the Node Selector API version setting.	v1	Allowed Values: • v1 • v2	NA
nodeSelector.nodeKey	Indicates the NodeSelector key configuration at the microservice level . When helmBasedConfigurationNodeSelectorApiVersion is set to v1, this configuration comes into use. This configuration does not depend on the nodeSelection flag. Once configured, this is used for all the microservices.	''	Valid key for a label for a particular node	NA
nodeSelector.nodeValue	Indicates the NodeSelector Value configuration at the global level . When helmBasedConfigurationNodeSelectorApiVersion is set to v1, this configuration comes into use. This configuration does not depend on the nodeSelection flag. Once configured, this is used for all the microservices.	''	Valid value pair for the above key for a label for a particular node	NA
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2. Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod	NA

Table 3-3 (Cont.) nudr-dr-provservice microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
convertUeidToStandard	Enable to convert ueId key to standard format while provisioning or updating profile.	false	true/false	NA
onDemandMigrationProvEnabled	Flag to enable or disable on-demand migration for provisioning interface	false	true or false	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	30s Unit: Seconds	NA	NA
provLogsEnabled	Flag to enable or disable provision logging feature.	false	true/false	NA
provLogsApiNames	List of API names supported for provision logging feature.	nudr-dr-prov	nudr-dr-prov,nudr-group-id-map-prov,slf-group-prov,n5g-eir-prov	NA
notificationThreadPool.corePoolSize	Thread pool configurations for notification processing. **Note**: This is engineering configurations, you must not change the default value.	100	NA	NA
notificationThreadPool.maxPoolSize	Thread pool configurations for notification processing. **Note**: This is engineering configurations, you must not change the default value.	200	NA	NA
notificationThreadPool.maxQueueCapacity	Thread pool configurations for notification processing. **Note**: This is engineering configurations, you must not change the default value.	5000	NA	NA

3.1.4 nudr-notify-service Microservice Parameters

Following table provides the parameters for nudr-notify-service microservice.

Table 3-4 nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
enabled	flag to enable nudr-notify-service.	true	true or false	For SLF deployment, this microservice must not be enabled.
image.name	Docker Image name.	nudr_notify_service	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
image.tag	Tag or version of Image.	24.3.0	NA	NA
image.pullPolicy	A value that defines whether to pull an image or not.	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
notification.retrycount	Number of notification attempts in case of notification failures.	2	Range: : 0 - 10	Retry is based on **notification.retryerrorcodes** configuration
notification.retryinterval	The retry interval for notifications in case of failure.	5 Unit: Seconds	Range: 1 - 60	NA
notification.retryerrorcodes	Error codes eligible for retry notifications in case of failures.	"400,429,500,503"	Valid HTTP status codes comma separated.	Give comma separated error codes.
notification.retryNotificationExpiry	It is the expiry time to delete the expired notifications.	86400 Unit: Seconds	NA	NA
notification.retryNotificationRecordsLimit	Maximum limit of notifications that can be stored on database for retry. If the limit are reached, the oldest retry notifications is deleted.	50000	Maximum limit: 50000	NA
notification.resourceRemovalNotificationEnabled	It is the notifications to delete subscriber and policy data. When the flag is set to true, it includes the "delResources" attribute and key in the notification payload. If the flag set to false, it will send a payload with policy data attributes with null values.	true	true or false	NA
hikari.poolsize	Mysql Connection pool size.	30	NA	Create the hikari pool connection size at start up
hikari.connectionTimeout	MYSQL connection timeout.	1000 Unit: Milliseconds	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
hikari.minimumIdleConnections	Minimum idle MYSQL connections maintained	2	NA	Valid. No, less than poolsize configured
hikari.idleTimeout	Idle MYSQL connections timeout.	10000 Unit: Milliseconds	NA	NA
hikari.queryTimeout	Query timeout for a single database query. If set to -1, there will be no timeout set for database queries.	-1 Unit: Seconds	NA	NA
serviceToCheck	Essential services on which init container performs readiness check.	nudr-drservice	Comma separated microservice names	NA
internalTracingEnabled	Flag to enable Jaeger tracing for nudr-notify-service.	false	true/false	NA
maxConcurrentPushedStreams	Maximum number of concurrent requests that can be pushed per destination.	6000	NA	NA
maxRequestsQueuedPerDestination	Maximum number of requests that can be queued per destination.	5000	NA	NA
maxConnectionsPerDestination	Maximum connection allowed per destination.	10	NA	NA
maxConnectionsPerIp	Maximum connection allowed per IP.	10	NA	NA
requestTimeout	Client request timeout.	5000 Unit: Milliseconds	NA	NA
connectionTimeout	Client connection timeout.	10000 Unit: Milliseconds	NA	NA
idleTimeout	Client connection idle timeout.	0 Unit: Milliseconds	NA	NA
dnsRefreshDelay	Server DNS refresh delay.	120000 Unit: Milliseconds	NA	NA
pingDelay	Server ping delay.	60 Unit: Milliseconds	NA	NA
connectionFailureThreshold	Client connection failure threshold.	10	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
jettyserviceMeshCheck	Flag to enable or disable jetty service mesh check for ASM deployment.	false	true or false	NA
logging.level.root	Log level of the notify service pod.	WARN	Possible Values: • WARN • INFO • DEBUG	NA
deployment.replicaCount	Number of nudr-notify-service pods to be maintained by replica set created with deployment.	2	NA	NA
minReplicas	Minimum number of pods.	2	NA	NA
maxReplicas	Maximum number of pods.	4	NA	NA
maxUnavailable	Number of replicas that can go down during a disruption.	1	NA	This parameter is used for PodDisruptionBudget Kubernetes resource
http.proxy.port	Port for connecting to Egress Gateway service.	*egressport	NA	Do not change the reference
service.type	Kubernetes service type for exposing UDR deployment.	ClusterIP	Possbile Values- ClusterIP NodePort LoadBalancer	Suggested to be set as ClusterIP (default value) always
service.port.http	The http port to be used in notify service to receive signals from nudr-notify-service pod.	5001	NA	NA
service.port.https	The https port to be used in notify service to receive signals from nudr-notify-service pod.	5002	NA	NA
service.port.management	The actuator management port to be used for notify service.	9000	NA	NA
resources.requests.cpu	Number of CPUs allocated for notify service pod during deployment.	2	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.requests.memory	Memory allocated for nudr-notify-service pod during deployment.	2Gi	NA	NA
resources.requests.logStorage	Log storage for ephemeral storage allocation request.	*containersLogStorageRequestsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.requests.crictlStorage	Crictl storage for ephemeral storage allocation request.	*containersCrictlStorageRequestsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.limits.cpu	Number of CPUs allocated during deployment.	2	NA	NA
resources.limits.memory	Memory allotment limitation	2Gi	NA	NA
resources.limits.logStorage	Log storage for ephemeral storage allocation limit.	*containersLogStorageLimitsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.limits.crictlStorage	Crictl storage for ephemeral storage allocation limit.	*containersCrictlStorageLimitsRef Unit: MB	NA	Change the reference to a different value if allocation needs to be different for this container
resources.target.averageCpuUtil	CPU utilization limit for autoscaling (creating HPA).	80	NA	NA
service.customExtension.labels	Custom Labels added to nudr-notify-service specific service.	null	NA	NA
service.customExtension.annotations	This can be used to add custom annotation(s) to nudr-notify-service.	null	NA	NA
deployment.customExtension.labels	This can be used to add custom label(s) to nudr-notify-service deployment.	null	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
deployment.customExtension.annotations	This can be used to add custom annotation(s) to nudr-notify-service deployment.	null	NA	NA
startupProbe.failureThreshold	It is the configurable number of times the startup probe is retried for failures. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	40 Unit: Seconds	NA	NA
startupProbe.periodSeconds	It is the time interval for every startup probe check. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	5 Unit: Seconds	NA	NA
extraContainers	This configuration decides service level extraContainer support. The default value is dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA
serviceMeshCheck	Enable when deployed in serviceMesh. Set it to false when side car is not included for this service	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference
istioSidecarReadyUrl	Readiness url configurable for side car Change only if the url is different for the side car container in this microservice	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level . If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

tolerations	When tolerationsSetting is ENABLED configure tolerations here.	[]	NA
			Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2	NA
nodeSelector.nodeKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag. Once configured this is used for all microservices.	' '	Valid key for a label for a particular node	NA
nodeSelector.nodeValue	NodeSelector value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid value pair for the above key for a label for a particular node	NA
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2 Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	30s Unit: Seconds	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
egressThreadPool.corePoolSize	Thread pool configurations for notification processing. It is the minimum number of threads used by the pool. **Note**: This is engineering configurations, you must not change the default value.	100	NA	NA
egressThreadPool.maxPoolSize	Thread pool configurations for notification processing. It is the maximum number of threads used by the pool. **Note**: This is engineering configurations, you must not change the default value.	200	NA	NA
egressThreadPool.maxQueueCapacity	Thread pool configurations for notification processing. It is the maximum thread a pool can hold. **Note**: This is engineering configurations, you must not change the default value.	5000	NA	NA
retryProcessThreadPool.corePoolSize	Thread pool configurations for retry and delete notifications processing. It is the minimum number of threads used by the pool. **Note**: This is engineering configurations, you must not change the default value.	100	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
retryProcessThreadPool.maxPoolSize	Thread pool configurations for retry and delete notifications processing. It is the maximum number of threads used by the pool. **Note**: This is engineering configurations, you must not change the default value.	200	NA	NA
retryProcessThreadPool.maxQueueCapacity	Thread pool configurations for retry and delete notifications processing. It is the maximum thread a pool can hold. **Note**: This is engineering configurations, you must not change the default value.	2000	NA	NA
cleanupThreadPool.corePoolSize	Thread pool configurations for retry and delete notifications processing. It is the minimum number of threads used by the pool. **Note**: This is engineering configurations, you must not change the default value.	10	NA	NA
cleanupThreadPool.maxPoolSize	Thread pool configurations for retry and delete notifications processing. It is the maximum number of threads used by the pool. **Note**: This is engineering configurations, you must not change the default value.	20	NA	NA

Table 3-4 (Cont.) nudr-notify-service microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
cleanupThreadPool.maxQueueCapacity	Thread pool configurations for retry and delete notifications processing. It is the maximum thread a pool can hold. **Note**: This is engineering configurations, you must not change the default value.	500	NA	NA

3.1.5 nrf-client-service Microservice Parameters

Based on UDR mode: configmapApplicationConfig.profile.appProfiles

If the alternate route service is enabled, then configure **alternateNRFRetryCount** to 1 in the NRF retry configuration as follows:

```
nrfRetryConfig=[{"serviceRequestType":"ALL_REQUESTS","primaryNRFRetryCount":1,
"nonPrimaryNRFRetryCount":1,"alternateNRFRetryCount":1,"errorReasonsForFailure
":
["503","504","500","SocketTimeoutException","JsonProcessingException","Unknown
HostException","NoRouteToHostException","IOException"],"gatewayErrorCodes":
["503"],"requestTimeout":10},
{"serviceRequestType":"AUTONOMOUS_NFREGISTER","primaryNRFRetryCount":1,"nonPri
maryNRFRetryCount":1,"alternateNRFRetryCount":-1,"errorReasonsForFailure":
["503","504","500","SocketTimeoutException","JsonProcessingException","Unknown
HostException","NoRouteToHostException","IOException"],"gatewayErrorCodes":
["503"],"requestTimeout":10}]
healthCheckConfig={"healthCheckCount":-1,"healthCheckInterval":5,"requestTimeo
ut":10,"errorReasonsForFailure":
["503","504","500","SocketTimeoutException","IOException"],"gatewayErrorCodes"
:[]}
```

There are four different profiles for appProfiles field as follows:

> **Note**
>
> InstanceId passed in profile should be same as that mentioned in global section.

Table 3-5 appProfiles Details

UDSDF ENABLED	MODE	PROFILE
FALSE	ALL	[{"nfInstanceId":"5a7bd676-ceeb-44bb-95e0-f6a55a328b03","nfStatus":"REGISTERED","fqdn":"ocudr-ingressgateway.myudr.svc.cluster.local","nfType":"UDR","allowedNfTypes":["PCF","UDM","NRF"],"plmnList":[{"mnc":"14","mcc":"310"}],"priority":10,"capacity":500,"load":0,"locality":"bangalore","nfServices":[{"load":0,"scheme":"http","versions":[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-ingressgateway.myudr.svc.cluster.local","ipEndPoints":[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus":"REGISTERED","allowedNfTypes":["PCF","UDM"],"serviceInstanceId":"ae870316-384d-458a-bd45-025c9e748976","serviceName":"nudr-dr","priority":10,"capacity":500},{"load":0,"scheme":"http","versions":[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-ingressgateway.myudr.svc.cluster.local","ipEndPoints":[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus":"REGISTERED","allowedNfTypes":["NRF"],"serviceInstanceId":"547d42af-628a-4d5d-a8bd-38c4ba672682","serviceName":"nudr-group-id-map","priority":10,"capacity":500}],"udrInfo":{"supportedDataSets":["POLICY","SUBSCRIPTION"],"groupId":"udr-1","externalGroupIdentifiersRanges":[{"start":"10000000000","end":"20000000000"}],"supiRanges":[{"start":"10000000000","end":"20000000000"}],"gpsiRanges":[{"start":"10000000000","end":"20000000000"}]},"heartBeatTimer":90,"nfServicePersistence":false,"nfProfileChangesSupportInd":false,"nfSetIdList":

Table 3-5 (Cont.) appProfiles Details

UDSDF ENABLED	MODE	PROFILE
		["setxyz.udrset.5gc.mnc012.mcc345"]}]
	NUDR-DR	[{"nfInstanceId":"5a7bd676-ceeb-44bb-95e0-f6a55a328b03","nfStatus":"REGISTERED","fqdn":"ocudr-ingressgateway.myudr.svc.cluster.local","nfType":"UDR","allowedNfTypes":["PCF","UDM","NRF"],"plmnList":[{"mnc":"14","mcc":"310"}],"priority":10,"capacity":500,"load":0,"locality":"bangalore","nfServices":[{"load":0,"scheme":"http","versions":[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-ingressgateway.myudr.svc.cluster.local","ipEndPoints":[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus":"REGISTERED","allowedNfTypes":["PCF","UDM"],"serviceInstanceId":"ae870316-384d-458a-bd45-025c9e748976","serviceName":"nudr-dr","priority":10,"capacity":500}],"udrInfo":{"supportedDataSets":["POLICY","SUBSCRIPTION"],"groupId":"udr-1","externalGroupIdentifiersRanges":[{"start":"10000000000","end":"20000000000"}],"supiRanges":[{"start":"10000000000","end":"20000000000"}],"gpsiRanges":[{"start":"10000000000","end":"20000000000"}]},"heartBeatTimer":90,"nfServicePersistence":false,"nfProfileChangesSupportInd":false,"nfSetIdList":["setxyz.udrset.5gc.mnc012.mcc345"]}]

Table 3-5 (Cont.) appProfiles Details

UDSDF ENABLED	MODE	PROFILE
	NUDR-GROUP-ID-MAP	[{"nfInstanceId":"5a7bd676-ceeb-44bb-95e0-f6a55a328b03","nfStatus":"REGISTERED","fqdn":"udr1-ingressgateway.atspatch-1111","nfType":"UDR","allowedNfTypes":["NRF"],"plmnList":[{"mnc":"14","mcc":"310"}],"priority":10,"capacity":500,"load":0,"locality":"bangalore","nfServices":[{"load":0,"scheme":"http","versions":[{"apiFullVersion":"2.1.0.alpha-3","apiVersionInUri":"v1"}],"fqdn":"ocudr-ingressgateway.myudr.svc.cluster.local","ipEndPoints":[{"port":"80","ipv4Address":"10.0.0.0","transport":"TCP"}],"nfServiceStatus":"REGISTERED","allowedNfTypes":["NRF"],"serviceInstanceId":"547d42af-628a-4d5d-a8bd-38c4ba672682","serviceName":"nudr-group-id-map","priority":10,"capacity":500}],"udrInfo":{"groupId":"udr-1","externalGroupIdentifiersRanges":[{"start":"10000000000","end":"20000000000"}],"supiRanges":[{"start":"10000000000","end":"20000000000"}],"gpsiRanges":[{"start":"10000000000","end":"20000000000"}]},"heartBeatTimer":90,"nfServicePersistence":false,"nfProfileChangesSupportInd":false,"nfSetIdList":["setxyz.udrset.5gc.mnc012.mcc345"]}]

Table 3-5 (Cont.) appProfiles Details

UDSDF ENABLED	MODE	PROFILE
	N5G-EIR-EIC (EIR Mode)	[{"nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03", "nfStatus": "REGISTERED", "fqdn": "ocudr-ingressgateway.myudr.svc.cluster.local", "nfType": "5G_EIR", "allowedNfTypes": ["AMF"], "plmnList": [{ "mnc": "14", "mcc": "310" }], "priority": 10, "capacity": 500, "load": 0, "locality": "bangalore", "nfServices": [{ "load": 0, "scheme": "http", "versions": [{ "apiFullVersion": "2.1.0.alpha-3", "apiVersionInUri": "v1" }], "fqdn": "ocudr-ingressgateway.myudr.svc.cluster.local", "ipEndPoints": [{ "port": "80", "ipv4Address": "10.0.0.0", "transport": "TCP" }], "nfServiceStatus": "REGISTERED", "allowedNfTypes": ["AMF"], "serviceInstanceId": "ae870316-384d-458a-bd45-025c9e748976", "serviceName": "n5g-eir-eic", "priority": 10, "capacity": 500 }], "heartBeatTimer": 90, "nfServicePersistence": false, "nfProfileChangesSupportInd": false, "nfSetIdList": ["setxyz.udrset.5gc.mnc012.mcc345"]}]

Table 3-5 (Cont.) appProfiles Details

UDSDF ENABLED	MODE	PROFILE
TRUE	NA	[{"nfInstanceId":"abae35e5-cc45-4016-8dd4-89598e5311b9","nfType":"UDSF","nfStatus":"REGISTERED","heartBeatTimer":60,"plmnList":[{"mnc":"14","mcc":"310"}],"fqdn":"ocudr-ingressgateway.myudr.svc.cluster.local","allowedNfTypes":["PCF","UDM"],"capacity":1000,"load":0,"priority":10,"udsfInfo":{"groupId":"udr-1","supiRanges":[{"start":"10000000000","end":"20000000000"}]},"nfServicePersistence":false,"nfServices":[{"serviceInstanceId":"b066742a-c270-4c91-b59c-23afe81fa715","serviceName":"nudsf-dr","nfServiceStatus":"REGISTERED","priority":10,"capacity":1000,"scheme":"http","versions":[{"apiVersionInUri":"v1","apiFullVersion":"1.1.0.0"}]}],"nfProfileChangesSupportInd":false}]

Other nrf-client-service configuration parameters are as follows:

Table 3-6 nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
configmapApplicationConfig.profile.primaryNrfApiRoot	Primary NRF Hostname and Port	nrfDeployName-nrf-1-nrf-simulator.nrfDeployNamespace-nrf-1.svc:5808	M	NA
configmapApplicationConfig.profile.secondaryNrfApiRoot	Secondary NRF Hostname and Port	-	NA	NA
configmapApplicationConfig.profile.nrfScheme	Scheme of primary and secondary NRF http or https	http	NA	http, https
configmapApplicationConfig.profile.retryAfterTime	Default downtime(in Duration) of an NRF detected to be unavailable.	PT120S	NA	NA
configmapApplicationConfig.profile.nrfClientType	The NfType of the NF registering	NRF	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
configmapApplicationConfig.profile.nrfClientSubscribeTypes	the NFType for which the NF wants to subscribe to the NRF	PCF	NA	NA
configmapApplicationConfig.profile.appProfiles	The NfProfile of the NF to be registered with NRF	[{}]	NA	NA
configmapApplicationConfig.profile.enableF3	Support for 29.510 Release 15.3	true	NA	NA
configmapApplicationConfig.profile.enableF5	Support for 29.510 Release 15.5	true	NA	NA
configmapApplicationConfig.profile.registrationRetryInterval	Retry Interval after a failed autonomous registration request	5000	NA	1.4.0
configmapApplicationConfig.profile.subscriptionRetryInterval	Retry Interval after a failed autonomous subscription request	5000	NA	1.4.0
configmapApplicationConfig.profile.discoveryRetryInterval	Retry Interval after a failed autonomous discovery request	5000	NA	1.4.0
configmapApplicationConfig.profile.renewalTimeBeforeExpiry	Time Period(seconds) before the Subscription Validity time expires	3600	NA	NA
configmapApplicationConfig.profile.validityTime	The default validity time(days) for subscriptions	30	NA	NA
configmapApplicationConfig.profile.enableSubscriptionAutoRenewal	Enable Renewal of Subscriptions automatically	true	NA	NA
configmapApplicationConfig.profile.nfHeartbeatRate	This value specifies the rate at which the NF shall heartbeat with the NRF. The value shall be configured in terms of percentage(1-100). if the heartbeatTimer is 60s, then the NF shall heartbeat at nfHeartBeatRate * 60/100	-	NA	NA
configmapApplicationConfig.profile.acceptAdditionalAttributes	Enable additionalAttributes as part of 29.510 Release 15.5	false	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
configmapApplicationConfig.profile.retryForCongestion	The duration(seconds) after which nrf-client should retry to a NRF server found to be congested	5	NA	NA
configmapApplicationConfig.profile.supportedDataSetId	The data-set value to be used in queryParams for NFs autonomous/on-demand discovery	-	NA	NA
configmapApplicationConfig.profile.enableVirtualNrfResolution	Enable virtual NRF session retry by Alternate routing service	false	NA	NA
configmapApplicationConfig.profile.virtualNrfFqdn	Virtual NRF FQDN used to query static list of route	nrf.oracle.com	NA	NA
configmapApplicationConfig.profile.virtualNrfScheme	http or https	http	NA	NA
configmapApplicationConfig.profile.virtualNrfPort	-	-	NA	1.4.0
configmapApplicationConfig.profile.useAlternateScpOnAlternateRouting	Enable use SCP on alternate routing service	-	NA	NA
configmapApplicationConfig.profile.subscriberNotificationRetry	Number of health status notification retries to a subscriber	2	NA	1.4.0
configmapApplicationConfig.profile.timeoutUrlPaths	Timeout Non 3GPP URL Resource Path(s) ConfigurationtimerProfileJson	-	NA	NA
configmapApplicationConfig.profile.timerProfileJson	Timer Profile JSON Configuration	-	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
configmapApplicationConfig.profile.requestTimeoutGracePeriod	The grace period at nrf-client for which it shall wait for a response from the NRF. This value shall be added to value configured at configmapApplicationConfig.profile.requestTimeout. Unit: seconds The setting support for 2s (means in seconds) or 50ms (means in milliseconds) format from Release 1.6.x	2	NA	1.4.1
configmapApplicationConfig.profile.nrfRetryConfig	Configurations required for the NRF Retry mechanism NRF Alternate Route Retry Feature	-	NA	1.4.0
configmapApplicationConfig.profile.serviceRequestType	Type of service operation for which configuration is applicable NRF Alternate Route Retry Feature	ALL_REQUESTS	NA	1.4.0
configmapApplicationConfig.profile.primaryNRFRetryCount	Number of retries for primary instance NRF Alternate Route Retry Feature	1	NA	1.4.0
configmapApplicationConfig.profile.nonPrimaryNRFRetryCount	Number of retries for non-primary instance NRF Alternate Route Retry Feature	1	NA	1.4.0
configmapApplicationConfig.profile.alternateNRFRetryCount	Number of retry attempt NRF Alternate Route Retry Feature	1	NA	1.4.0
configmapApplicationConfig.profile.errorReasonsForFailure	httpStatusCode or error conditions for which retry shall be attempted NRF Alternate Route Retry Feature	["503","504","500,"SocketTimeoutException","JsonProcessingException","UnknownHostException","NoRouteToHostException","IOException""]	NA	1.4.1
configmapApplicationConfig.profile.requestTimeout	Timeout period(seconds) where no response is received from NRF NRF Alternate Route Retry Feature	10	NA	1.4.0

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
configmapApplicationConfig.profile.gatewayErrorCodes	httpStatusCode sent by gateway for which retry shall be attempted NRF Alternate Route Retry Feature	["503"]	NA	1.4.0
configmapApplicationConfig.profile.healthCheckConfig	Configurations required for the Health check of NRFs NRF Alternate Route Retry Feature	-	NA	1.4.0
configmapApplicationConfig.profile.healthCheckCount	Consecutive success/ failure operation count (default -1 means disabled) NRF Alternate Route Retry Feature	1	NA	1.4.0
configmapApplicationConfig.profile.healthCheckInterval	Interval duration in seconds to perform health check NRF Alternate Route Retry Feature	5	NA	1.4.0
configmapApplicationConfig.profile.requestTimeout	Timeout period(seconds) where no response is received from NRF NRF Alternate Route Retry Feature	10	NA	1.4.0
configmapApplicationConfig.profile.errorReasonsForFailure	httpStatusCode or error conditions for which the NRF is considered as unhealthy NRF Alternate Route Retry Feature	["503","504","500,"SocketTimeoutException","JsonProcessingException","UnknownHostException","NoRouteToHostException","IOException""]	NA	1.4.1
configmapApplicationConfig.profile.gatewayErrorCodes	httpStatusCode sent by gateway for the NRF shall be considered unhealthy NRF Alternate Route Retry Feature	[]	NA	1.4.0
configmapApplicationConfig.profile.enableNrfRetry	enable NRF retry	-	NA	1.4.1
configmapApplicationConfig.profile.maxNrfRetries	Number of retry attempt	-	NA	1.4.1
configmapApplicationConfig.profile.enableNrfAlternateRouting	enable NRF alternate routing service	-	NA	1.4.1
configmapApplicationConfig.profile.alternateRoutingErrorCodes	set alternate routing error codes	-	NA	1.4.1

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
nrf-client-nfdiscovery	-	-	NA	NA
configmapApplicationConfig	A reference to the nrf-client.configMapApplication	*configRef	NA	NA
nfProfileConfigMode	Configuration Mode for NfProfilevalues: - REST, HELM	REST	NA	NA
image	NRF Client Microservice image name	nrf-client	NA	NA
imageTag	NRF Client Microservice image tag	24.3.2	NA	NA
envJaegerSamplerParam	-	1	NA	NA
envJaegerSamplerType	-	ratelimiting	NA	NA
envJaegerServiceName	-	nrf-client-nfdiscovery	NA	NA
cpuRequest	Resource Details	2	NA	NA
cpuLimit	-	2	NA	NA
memoryRequest	-	1Gi	NA	NA
memoryLimit	-	1Gi	NA	NA
minReplicas	Min replicas to scale to maintain an average CPU utilization	2	NA	NA
maxReplicas	Max replicas to scale to maintain an average CPU utilization	5	NA	NA
averageCpuUtil	-	80	NA	NA
type	-	ClusterIP	NA	NA
cacheDiscoveryResults	Set to true if the discovery results should be cached.	true	NA	NA
envDiscoveryServicePort	Discovery Service Port used for subscribing to management Service	5910	NA	NA
envManagementServicePort	Management Service Port used to send subscriptions to the Management Service	5910	NA	NA
commonCfgClient.enabled	Flag to enable/disable dynamic logging using common configuration service	false	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
commonCfgServer.configServerSvcName	This attribute shall be configured only if commonCfgClient.enabled is set to true.The Nf Configuration Service name or the common config service. Do not comment this line in case you want to deploy both config-server and APIGW in same namespace simultaneously Otherwise comment this line and use 'host'	'common-config-server'	NA	NA
commonCfgServer.host	The Host name of the Nf Configuration Service name or the common config service.This attribute shall be configured only if commonCfgClient.enabled is set to true.	'common-config-server'	NA	NA
commonCfgServer.port	The port of the Nf Configuration Service name or the common config service.This attribute shall be configured only if commonCfgClient.enabled is set to true.	80	NA	NA
commonCfgServer.pollingInterval	The interval at which the discovery service shall poll the configuration service to check for updates in msThis attribute shall be configured only if commonCfgClient.enabled is set to true.	5000	NA	NA
dbConfig.dbHost	The db host to which the helm hooks shall connect to load the json scheme and default configurations.This attribute shall be configured only if commonCfgClient.enabled is set to true.	10.75.225.4	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
dbConfig.dbPort	The db port to which the helm hooks shall connect to load the json scheme and default configurations.This attribute shall be configured only if commonCfgClient.enabled is set to true.	3306	NA	NA
dbConfig.secretName	The database secretThis attribute shall be configured only if commonCfgClient.enabled is set to true.	'nrfclient-mysql'	NA	NA
dbConfig.dbName	The database name which will be used to store the common configuraiton.This attribute shall be configured only if commonCfgClient.enabled is set to true.	ocpm_config_server	NA	NA
dbConfig.dbUNameLiteral	The db literal name that shall be used as per the secrets configured.This attribute shall be configured only if commonCfgClient.enabled is set to true.	mysql-username	NA	NA
dbConfig.dbPwdLiteral	The db password literal name that shall be used as per the secrets configured. This attribute shall be configured only if commonCfgClient.enabled is set to true.	mysql-password	NA	NA
logging.level.root	default system log level	WARN	NA	NA
logging.level.nrfclient	default application log level	WARN	NA	NA
serviceMeshCheck	This attributes allows to enable serviceMesh when used with ASM deployment.	false	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
istioSidecarQuitUrl	The sidecar (istio quit url) when deployed in serviceMesh. This value shall be considered only when serviceMeshCheck is true. default: http://127.0.0.1:15000/quitquitquit	""	NA	NA
istioSidecarReadyUrl	The sidecar (istio ready url) when deployed in serviceMesh. This value shall be considered only when serviceMeshCheck is true.default :http://127.0.0.1:15000/ready	""	NA	NA
metricPrefix	A prefix that shall be added to all the metric names.By default, this shall contain the value configured in the global section metricPrefix	"oc"	NA	NA
metricSuffix	A suffix that shall be added to all the metric names.By default, this shall contain the value configured in the global section metricSuffix	""	NA	NA
cacheServicePortStart	Coherence Port for communication	8095	NA	NA
cacheServicePortEnd	Coherence Port for communication	8096	NA	NA
nrf-client-nfmanagement	-	-	NA	NA
configmapApplicationConfig	A reference to the nrf-client.configMapApplication	*configRef	NA	NA
nfProfileConfigMode	Configuration Mode for NfProfile values: - REST, HELM	REST	NA	NA
image	Deployment specific configuration for Nrf-Client Management Microservice. NRF Client Microservice image name.	nrf-client	NA	NA
imageTag	NRF Client Microservice image tag	24.3.2	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
envJaegerSamplerParam	-	'1'	NA	NA
envJaegerSamplerType	-	ratelimiting	NA	NA
envJaegerServiceName	-	nrf-client-nfmanagement	NA	NA
replicas	Resource Details	1	NA	NA
cpuRequest	CPU Request	1	NA	NA
cpuLimit	CPU Limit	1	NA	NA
memoryRequest	Memory Request	1Gi	NA	NA
memoryLimit	Memory Limit	1Gi	NA	NA
type		ClusterIP	NA	NA
commonCfgClient.enabled	Flag to enable/disable dynamic logging using common configuration service	false	NA	NA
commonCfgServer.configServerSvcName	This attribute shall be configured only if commonCfgClient.enabled is set to true.The Nf Configuration Service name or the common config service. Do not comment this line in case you want to deploy both config-server and APIGW in same namespace simultaneously Otherwise comment this line and use 'host'	'common-config-server'	NA	NA
commonCfgServer.host	The Host name of the Nf Configuration Service name or the common config service.This attribute shall be configured only if commonCfgClient.enabled is set to true.	'common-config-server'	NA	NA
commonCfgServer.port	The port of the Nf Configuration Service name or the common config service.This attribute shall be configured only if commonCfgClient.enabled is set to true.	80	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
commonCfgServer.pollingInterval	The interval at which the discovery service shall poll the configuration service to check for updates in msThis attribute shall be configured only if commonCfgClient.enabled is set to true.	5000	NA	NA
dbConfig.dbHost	The db host to which the helm hooks shall connect to load the json scheme and default configurations.This attribute shall be configured only if commonCfgClient.enabled is set to true.	10.75.225.4	NA	NA
dbConfig.dbPort	The db port to which the helm hooks shall connect to load the json scheme and default configurations.This attribute shall be configured only if commonCfgClient.enabled is set to true.	3306	NA	NA
dbConfig.secretName	The database secretThis attribute shall be configured only if commonCfgClient.enabled is set to true.	nrfclient-mysql	NA	NA
dbConfig.dbName	The database name which will be used to store the common configuraiton.This attribute shall be configured only if commonCfgClient.enabled is set to true.	ocpm_config_server	NA	NA
dbConfig.dbUNameLiteral	The db literal name that shall be used as per the secrets configured.This attribute shall be configured only if commonCfgClient.enabled is set to true.	mysql-username	NA	NA

Table 3-6 (Cont.) nrf-client-service parameters

Parameter	Description	Default Value	M/O/C	Ranges or Possible Values
dbConfig.dbPwdLiteral	The db password literal name that shall be used as per the secrets configuredThis attribute shall be configured only if commonCfgClient.enabled is set to true.	mysql-password	NA	NA
logging.level.root	default system log level	WARN	NA	NA
logging.level.nrfclient	default application log level	WARN	NA	NA
serviceMeshCheck	This attributes allows to enable serviceMesh when used with ASM deployment.	false	NA	NA
istioSidecarQuitUrl	The sidecar (istio quit url) when deployed in serviceMesh. This value shall be considered only when serviceMeshCheck is true.	""	NA	NA
istioSidecarReadyUrl	The sidecar (istio ready url) when deployed in serviceMesh. This value shall be considered only when serviceMeshCheck is true.default	""	NA	NA
metricPrefix	A prefix that shall be added to all the metric names.By default, this shall contain the value configured in the global section metricPrefix	"oc"	NA	NA
metricSuffix	A suffix that shall be added to all the metric names.By default, this shall contain the value configured in the global section metricSuffix	""	NA	NA
cacheServicePortStart	Coherence Port for communication	8095	NA	NA
cacheServicePortEnd	Coherence Port for communication	8096	NA	NA

3.1.6 nudr-config Microservice Parameters

Following table provides the parameters for nudr-config microservice.

Table 3-7 nudr-config microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
enabled	Flag to enable nudr-config service	*configServerEnabled	NA	Do not change this parameter value
logging.level.root	Log level of the nudr-config pod	WARN	Possible Values: WARN INFO DEBUG	NA
image.name	Docker Image name	nudr_config	NA	NA
service.customExtension.labels	Use this to add custom label(s) to nudr-config service	null	NA	NA
service.customExtension.annotations	Use this to add custom annotation(s) to nudr-config service	null	NA	NA
deployment.customExtension.labels	Use this to add custom label(s) to nudr-config deployment	null	NA	NA

Table 3-7 (Cont.) nudr-config microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	Use this to add custom annotation(s) to nudr-config deployment	`annotations:`
		
		`sidecar.istio.io/proxyCPU: "1000m"`
		
		`sidecar.istio.io/proxyCPULimit: "1000m"`
		
		`sidecar.istio.io/proxyMemory: "1Gi"`
		
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		
		`proxy.istio.io/config: |`
		
		`terminationDrainDuration: 60s`
			NA	NA

hikari.poolsize	Configuration of MYSQL connection pool size	25	NA	NA
hikari.connectionTimeout	MYSQL connection timeout	1000	Unit: Milliseconds	NA
hikari.minimumIdleConnections	Minimum idle MYSQL connections maintained	2	Valid. No, less than poolsize configured	NA
hikari.idleTimeout	Idle MYSQL connections timeout	10000 Unit: Milliseconds	NA	NA
hikari.queryTimeout	Query timeout for a single database query. If set to -1, there will be no timeout set for database queries.	-1 Unit: Seconds	NA	NA

Table 3-7 (Cont.) nudr-config microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.type	Kubernetes service type for exposing UDR deployment	ClusterIP	Possible Values: ClusterIP NodePort LoadBalancer	Suggested to be set as ClusterIP (default value) always
image.pullPolicy	This setting ensures if image need to be pulled or not	IfNotPresent	Possible Values: Always IfNotPresent Never	NA
service.port.management	Actuator management port to be used for nudr-config service	9000	NA	NA
service.port.https	The https port to be used for nudr-config service	5002	NA	NA
service.port.http	The http port to be used in nudr-config service	5001	NA	NA
resources.target.averageCpuUtil	CPU utilization limit for autoscaling (creating HPA)	80	NA	NA
resources.requests.memory	Memory allocated for nudr-config pod during deployment	2Gi	NA	NA
resources.requests.logStorage	Log storage for ephemeral storage allocation request	*containersLogStorageRequestsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.requests.crictlStorage	Crictl storage for ephemeral storage allocation request	*containersCrictlStorageRequestsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.limits.memory	Memory allotment limit	2Gi	NA	NA
resources.requests.cpu	CPUs allocated for nudr-config pod during deployment	2	NA	NA
resources.limits.cpu	CPU allotment limitation	2	NA	NA
resources.limits.logStorage	Log storage for ephemeral storage allocation limit	*containersLogStorageLimitsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container

Table 3-7 (Cont.) nudr-config microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.limits.crictlStorage	Critical storage for ephemeral storage allocation limit	*containersCrictlStorageLimitsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
image.tag	Tag or version of Image	24.3.0	NA	NA
deployment.replicaCount	Number of nudr-config pods to be maintained by replica set created with deployment	2	NA	NA
minReplicas	Minimum number of pods	2	NA	NA
maxReplicas	Maximum number of pods	2	NA	NA
startupProbe.failureThreshold	It is the configurable number of times the startup probe is retried for failures. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	40	Unit: Seconds	
startupProbe.periodSeconds	It is the time interval for every startup probe check. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	5	Unit: Seconds	
extraContainers	This configuration decides service level extraContainer support. The default value is dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA

Table 3-7 (Cont.) nudr-config microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
serviceMeshCheck	Enable when deployed in serviceMesh. Set it to false when side car is not included for this service	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference
istioSidecarReadyUrl	Readiness url configurable for side car Change only if the url is different for the side car container in this microservice	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

Table 3-7 (Cont.) nudr-config microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

tolerations	When tolerationsSetting is ENABLED, configure tolerations here.	[]	NA
			Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2	NA
nodeSelector.nodeKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	''	Valid key for a label for a particular node	NA
nodeSelector.nodeValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	''	Valid value pair for the above key for a label for a particular node	NA

Table 3-7 (Cont.) nudr-config microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2. Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	30s	Unit: Seconds	NA

3.1.7 nudr-config-server Microservice Parameters

Following table provides the parameters for nudr-config-server Microservice.

Table 3-8 nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
enabled	Flag to enable nudr-config-server service	*configServerEnabled	NA	Do not change this default value
image	Image name of config-server	ocpm_config_server	NA	NA
imageTag	Tag or version of Image	24.3.3	NA	NA
pullPolicy	It helps to decide on policy for image pull	IfNotPresent	Allowed Values: IfNotPresent Never Always	NA
global.nfName	NF name used to add with config server service name	nudr	NA	NA
global.envJaegerAgentHost	Host FQDN for Jaeger agent service for config-server tracing	' '	NA	NA
global.envJaegerAgentPort	Port to connect to Jaeger agent for config-server tracing	6831	Valid Port	NA

Table 3-8 (Cont.) nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.serviceMeshEnabled	Enable when deployed in serviceMesh. Set this parameter to false when side car is not included for this service	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference
global.istioSidecarQuitUrl	Readiness url configurable for side car Change only if URL is different for side car container in this microservice	*istioQuitUrl	NA	Do not change this value unless it is different from the value configured in global section reference
global.istioSidecarReadyUrl	Readiness url configurable for side car Change this parameter value only if the url is different for side car container in this microservice	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference
global.logStorage	Log storage for ephemeral storage request	*containersLogStorageRequestsRef	Unit MB	Do not change reference unless a different value needs to be configured
global.crictlStorage	Crictl storage for ephemeral storage request	*containersCrictlStorageRequestsRef	Unit MB	Do not change reference unless a different value needs to be configured
envLoggingLevelApp	Log level of the nudr-config-server pod	WARN	Possible Values: WARN INFO DEBUG	NA
replicas	Number of nudr-config-server pods to be maintained by replica set created with deployment	1	NA	NA
minAvailable	Number of pods that must be available, even during a disruption	0	NA	This parameter is used for PodDisruptionBudget Kubernetes resource

Table 3-8 (Cont.) nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.type	Kubernetes service type for exposing UDR deployment	ClusterIP	Possible Values: ClusterIP NodePort LoadBalancer	Suggested to be set as ClusterIP (default value) always
resources.requests.cpu	Number of CPUs allocated for nudr-config-server pod during deployment	2	NA	NA
resources.requests.memory	Memory allocated for nudr-config-server pod during deployment	512Mi	NA	NA
resources.limits.cpu	CPU allotment limitation	2	NA	NA
resources.limits.memory	Memory allotment limitation	2Gi	NA	NA
resources.limits.ephemeralStorage	Ephemeral storage allocation limits	1024Mi	Unit MB	NA
service.customExtension.labels	Use this parameter to add custom label(s) to nudr-config-server service	null	NA	NA
service.customExtension.annotations	Use this parameter to add custom annotation(s) to nudr-config-server service	null	NA	NA
deployment.customExtension.labels	Use this parameter to add custom label(s) to nudr-config-server deployment	null	NA	NA

Table 3-8 (Cont.) nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	Use this parameter to add custom annotation(s) to nudr-config-server deployment	`annotations:`
		
		`sidecar.istio.io/proxyCPU: "1000m"`
		
		`sidecar.istio.io/proxyCPULimit: "1000m"`
		
		`sidecar.istio.io/proxyMemory: "1Gi"`
		
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		
		`proxy.istio.io/config: |`
		
		`terminationDrainDuration: 60s`
			NA	NA

fullnameOverride	Full name to be used for configuration server service	udr-config-server	NA	This configuration must be updated to <Helm-release-name>-config-server if multiple ocudr/ocslf deployments are done within the same kubernetes namespace. Example: ocudr1-config-server
installedChartVersion	Chart version to be read by hooks	''	Valid version	NA

Table 3-8 (Cont.) nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
readinessProbe.initialDelaySeconds	Configurable wait time before performing the first readiness probe by Kubelet	70	NA Unit: Seconds	Do not change this value. If there is any delay in pod to come up and probe is killing the pod then you should tune these parameters
readinessProbe.periodSeconds	Time interval for every readiness probe check	10	NA Unit: Seconds	Do not change this value. If there is any delay in pod to come up and probe is killing the pod then you should tune these parameters
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	NA	Do not change this default value
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	1	NA	Do not change this default value
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes tries failureThreshold times before giving up	3	NA	Do not change this default value
livenessProbe.initialDelaySeconds	Configurable wait time before performing the first liveness probe by Kubelet	60	NA Unit: Seconds	Do not change this value. If there is any delay in pod to come up and probe is killing the pod then you should tune these parameters
livenessProbe.periodSeconds	Time interval for every liveness probe check	15	NA Unit: Seconds	Do not change this value. If there is any delay in pod to come up and probe is killing the pod then you should tune these parameters
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	NA	Do not change this default value

Table 3-8 (Cont.) nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	1	NA	Do not change this default value
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes tries failureThreshold times before giving up	3	NA	Do not change this default value
extraContainers	This configuration decides service level extraContainer support. The default value is dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

Table 3-8 (Cont.) nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

tolerations	When tolerationsSetting is ENABLED configure tolerations here.	[]	Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2	NA
nodeSelectorEnabled	Flag to enable v1 nodeSelector	false	true/false	NA

Table 3-8 (Cont.) nudr-config-server microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelectorKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid key for a label for a particular node	NA
nodeSelectorValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid value pair for the above key for a label for a particular node	NA
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2. Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	30s	Unit: Seconds	NA

3.1.8 nudr-diameterproxy Microservice Parameters

Following table list parameters for nudr-diameterproxy microservice.

Table 3-9 nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
enabled	Flag to enable nudr-diameterproxy service	true	NA	NA
image.name	Name of the Docker Image	nudr_diameterproxy	NA	NA
image.tag	Tag or version of Image	24.3.0	NA	NA
image.pullPolicy	This setting indicates if image need to be pulled or not	IfNotPresent	Possible Values: Always IfNotPresent Never	NA
logging.level.root	The log level of the nudr-diameterproxy server pod	WARN	Possible Values: WARN INFO DEBUG	NA
deployment.replicaCount	Number of nudr-config-server pods to be maintained by replica set created with deployment	2	NA	NA
serviceToCheck	Essential services on which init container may perform readiness checks	nudr-config-server	Comma separated microservice names	NA
minReplicas	Minimum number of pods of nudr-diameterproxy	2	NA	NA
maxReplicas	Maximum number of pods of nudr-diameterproxy	4	NA	NA
maxUnavailable	Number of replicas that can go down during a disruption	1	NA	This parameter is used for PodDisruptionBudget k8s resource
hikari.poolsize	Hikari pool connection size to be created at start up	20	NA	NA
hikari.connectionTimeout	MYSQL connection timeout	1000	Unit: Milliseconds	NA
hikari.minimumIdleConnections	Minimum idle MYSQL connections maintained	2	Valid. No, less than poolsize configured	NA
hikari.idleTimeout	Idle MYSQL connections timeout	10000	Unit: Milliseconds	NA

Table 3-9 (Cont.) nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
hikari.queryTimeout	Query timeout for a single database query. If set to -1, there will be no timeout set for database queries.	-1	Unit: Seconds	NA
maxConcurrentPushedStreams	Maximum number of concurrent requests that can be pushed per destination	6000	NA	NA
maxRequestsQueuedPerDestination	Maximum number of requests that can be queued per destination	5000	NA	NA
maxConnectionsPerDestination	Maximum connection allowed per destination	10	NA	NA
maxConnectionsPerIp	Maximum connection allowed per IP	10	NA	NA
requestTimeout	Client request timeout	5000	Unit: Milliseconds	NA
connectionTimeout	Client connection timeout	10000	Unit: Milliseconds	NA
idleTimeout	Client connection idle timeout	0	Unit: Milliseconds	NA
dnsRefreshDelay	Server DNS refresh delay	120000	Unit: Milliseconds	NA
pingDelay	Server ping delay	60	Unit: Milliseconds	NA
connectionFailureThreshold	Client connection failure threshold	10	NA	NA
jettyserviceMeshCheck	Flag to enable or disable jetty service mesh check for ASM deployment	false	true or false	NA
service.type	Kubernetes service type for exposing UDR deployment	ClusterIP	Possible Values: • ClusterIP • NodePort • LoadBalancer	Suggested to be set as ClusterIP (default value) always
service.port.http	The HTTP port to be used in nudr-diameterproxy service	5001	NA	NA
service.port.https	The HTTPS port to be used for nudr-diameterproxy service	5002	NA	NA

Table 3-9 (Cont.) nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.port.management	The actuator management port to be used for nudr-diameterproxy service	9000	NA	NA
service.port.diameter	The diameter port to be used for nudr-diameterproxy service	6000	NA	NA
resources.requests.cpu	The number of CPUs allocated for nudr-diameterproxy pod during deployment	2	NA	NA
resources.requests.memory	The memory to be allocated for nudr-diameterproxy pod during deployment	2Gi	NA	NA
resources.requests.logStorage	Log storage for ephemeral storage allocation request	*containersLogStorageRequestsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.requests.crictlStorage	Crictl storage for ephemeral storage allocation request	*containersCrictlStorageRequestsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.limits.cpu	Maximum number of CPUs allocated for nudr-diameterproxy pod	2	NA	NA
resources.limits.memory	Maximum memory to be allocated for nudr-diameterproxy pod	2Gi	NA	NA
resources.limits.logStorage	Log storage for ephemeral storage allocation limit	*containersLogStorageLimitsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container
resources.limits.crictlStorage	Crictl storage for ephemeral storage allocation limit	*containersCrictlStorageLimitsRef	Unit: MB	Change the reference to a different value if allocation needs to be different for this container

Table 3-9 (Cont.) nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.target.averageCpuUtil	CPU utilization limit for auto scaling (creating HPA)	80	NA	NA
drservice.port.http	HTTP port on which dr-service is running	5001	NA	NA
drservice.port.https	HTTPS port on which dr-service is running	5002	NA	dr-service port is required in diameterproxy application
diameter.realm	Host realm of diameterproxy	oracle.com	String value	NA
diameter.identity	FQDN (identity) of the diameterproxy in diameter messages.	nudr.oracle.com	String value	NA
diameter.strictParsing	Enable strict parsing of Diameter AVP and Messages.	false	NA	NA
diameter.IO.threadCount	Number of threads to handle IO operations in diameterproxy pod.	12	0 to 2* CPU	If the threadcount is 0 then application chooses the threadCount based on pod profile size
diameter.IO.queueSize	Queue size for IO (Input and output)	8192	2048 to 8192	Count should be the power of 2. If the queueSize is 0 then application chooses the queueSize based on pod profile size
diameter.messageBuffer.threadCount	Number of threads to handle messages in diameterproxy pod.	24	0 to 2* CPU	If the threadCount is 0, then application chooses the threadCount based on pod profile size.
diameter.messageBuffer.queueSize	It is the queue size of the processing message.	4096	Range: 1024 to 4096	NA
service.customExtension.labels	Use this to add custom label(s) to nudr-diameterproxy service.	null	NA	NA
service.customExtension.annotations	Use this to add custom annotation(s) to nudr-diameterproxy service.	null	NA	NA

Table 3-9 (Cont.) nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
deployment.customExtension.labels	Use this to add custom label(s) to nudr-diameterproxy deployment.	null	NA	NA
deployment.customExtension.annotations	Use this to add custom annotation(s) to nudr-diameterproxy deployment	null	NA	NA
startupProbe.failureThreshold	It is the configurable number of times the startup probe is retried for failures. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	40	Unit: Seconds	NA
startupProbe.periodSeconds	It is the time interval for every startup probe check. Note: Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters	5	Unit: Seconds	NA
extraContainers	This configuration decides service level extraContainer support. The default value lets it to be dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA
serviceMeshCheck	Flag to enable serviceMesh. Set this to false when side car is not included for this service	*serviceMeshFlag	NA	Do not change this value unless it is different from the value configured in global section reference

Table 3-9 (Cont.) nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
istioSidecarReadyUrl	Readiness URL configurable for side car Change this value only if the URL is different for side car container in this microservice	*istioReadyUrl	NA	Do not change this value unless it is different from the value configured in global section reference
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

Table 3-9 (Cont.) nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

tolerations	When tolerationsSetting is ENABLED, configure tolerations here.	[]	Example:
			
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2	NA
nodeSelector.nodeKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	''	Valid key for a label for a particular node	NA
nodeSelector.nodeValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	''	Valid value pair for the above key for a label for a particular node	NA

Table 3-9 (Cont.) nudr-diameterproxy microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2 Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod	NA
convertFieldsUnderSubscriberEntityToUpperCase	To convert the fields in the subscriber entity to uppercase	false	NA	NA
onDemandMigrationDiameterEnabled	Flag to enable or disable on-demand migration for diameter interface.	false	true or false	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit	30s	true or false	NA
convertFieldsUnderSubscriberEntityToUpperCase	Convert keys under subscriber entity to UpperCase	false	true or false	NA

3.1.9 appinfo microservice parameters

Following table list configurable parameters for appinfo microservice.

Table 3-10 appinfo microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
replicaCount	Indicates the number of pods, which needs to be created as part of deployment.	2	Valid number	NA
image	Name of the docker image of app info service	app-info	NA	NA
imageTag	Tag of the image of app info service	24.3.0	NA	NA
pullPolicy	Indicates if the image needs to be pulled or not.	IfNotPresent	Possible Values - Always IfNotPresent Never	NA

Table 3-10 (Cont.) appinfo microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
dbHookImage.name	Image name of dbHook	common_config_hook	Not Applicable	NA
dbHookImage.tag	Image tag name of dbHook	24.3.0	Not Applicable	NA
dbHookImage.pullPolicy	Indicates if the image needs to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
service.type	Defines service type used for UDR deployment	ClusterIP	Possible Values- ClusterIP NodePort LoadBalancer	NA
global.logStorage	Log storage for ephemeral storage request	*containersLogStorageRequestsRef	Unit MB	Do not change reference unless a different value needs to be configured
global.crictlStorage	Crictl storage for ephemeral storage request	*containersCrictlStorageRequestsRef	Unit MB	Do not change reference unless a different value needs to be configured
resources.limits.cpu	CPU Limit for app info pod	0.5	NA	NA
resources.limits.memory	Memory Limit for app info pod	1Gi	NA	NA
resources.limits.ephemeralStorage	Ephemeral storage allocation limits	1024Mi	Unit MB	NA
resources.requests.cpu	Requested CPU usage for app info pod to come up	0.5	NA	NA
resources.requests.memory	Requested memory usage for app info pod to come up	1Gi	NA	NA
nodeSelector	Node selector to stick to one slave node	{}	NA	NA
tolerations	Pod tolerations	[]	NA	NA
commonCfgClient.enabled	Set it to true if persistent configuration needs to be enabled.	true	true or false	NA

Table 3-10 (Cont.) appinfo microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
commonCfgServer.configServerSvcName	Service name of common configuration service to which the client tries to poll for configuration updates	nudr-config	Valid server service name. Example: nudr-config	NA
commonCfgServer.host	Host name of common configuration server to which client tries to poll for configuration updates. This value is picked up if commonCfgServer.configServerSvcName is not available.	10.75.224.123	Valid host details	NA
commonCfgServer.port	Port of common configuration server	5001	Valid Port	NA
commonCfgServer.pollingInterval	This is the interval between two subsequent polling requests from configuration client to server	5000	Valid period	NA
appinfo.debug	Specifies the log level of app info service. When this parameter is set to true, the log level of app info is DEBUG, otherwise, log level is INFO or WARN.	false	Possible Values - true or false	NA
log.level.appinfo	Identifies log level of app info	INFO	Possible Values: WARN DEBUG INFO	NA

Table 3-10 (Cont.) appinfo microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
infraServices	Infrastructure services. If using OCCNE 1.4 or if you don't want to monitor infra services such as db-monitor service, then the parameter should be set as an empty array. If infraServices is not set, by default appinfo shall monitor status of db-monitor-svc and db-replication-svc.	[]	NA	NA
core_services	List of microservices monitored by app info depending on UDR mode. It refers to *allordr or *slf or *eir from the Global Configurable Parameters section.	*allordr	Possible Values: *allordr, *slf, or 5g_eir: *eir	NA
istioSidecarQuitUrl	Defines the URL that is used for quitting service mesh side car. This URL is used to quit the istio side car after successful completion of hook job.	*istioQuitUrl	Not Applicable	Do not change this value unless the URL is different for the side car container in this microservice.
istioSidecarReadyUrl	Defines the URL that is used for checking the service mesh sidecar status and start application when the status is ready.	*istioReadyUrl	Not Applicable	Do not change this value unless the URL is different from the sidecar container in this microservice.
serviceMeshCheck	Set this parameter as true, when side car is deployed in serviceMesh and as false when side car is not included for this service.	*serviceMeshFlag	Not Applicable	Do not change this value unless it is different from the value configured in the global section

Table 3-10 (Cont.) appinfo microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

tolerations	When tolerationsSetting is ENABLED, configure tolerations here.	[]	Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v2	Allowed Values: • v1 • v2	NA

Table 3-10 (Cont.) appinfo microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2 For v1, you can have key:vaue pair configured.	{}	Valid key value pair matching a node for nodeSelection by a pod v1 Example: zone: Antartica	NA
maxUnavailable	Number of replicas that can go down during a disruption	50%	Not applicable	NA
watchMySQL	It monitors the MySQL connectiviy service	false	true or false	NA
dbStatusUri	URI of the MySQL connector database monitor service. URL must be changed according to the setup	`http://occne-db-monitor-svc.occne-infra:8080/db-tier/status/local`	NA	NA
scrapeInterval	Time interval in seconds, the app try to connect to database.	5	NA	NA
replicationStatusCheck	Enabling the flag will monitor the replication status	false	true or false	NA
realtimeDbStatusUri	This will monitor the realtime database status of the URL	`http://occne-db-monitor-svc.occne-infra:8080/db-tier/status/cluster/local/realtime`	NA	NA
replicationUri	This will monitor the replication status from app to the database in site	`http://occne-db-monitor-svc.occne-infra:8080/db-tier/status/replication`	NA	NA
replicationInterval	This is the configuration to scrape and monitor the replication status from app to database	30	NA	NA

Table 3-10 (Cont.) appinfo microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
prometheusUrl	This is the promethues service URL. The URL must be changed according to the setup.	`http://occne-kube-prom-stack-kube-prometheus.occne-infra:80/cne-23-1/prometheus`	NA	NA
alertmanagerUrl	This is the alert manager service URL. The URL must be changed according to the setup.	`http://occne-prometheus-alertmanager.occne-infra:80`	NA	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	60s	Unit: Seconds	NA
controlled_shutdown.operationlStateResourcePath	This is control shutdown feature operational state information retrieval path.	"/operationalState"	NA	NA
global.nfType	NFType used by app-info service	udr	Valid nftype, should be small letters	NA

3.1.10 nudr-perf-info Microservice Parameters

Following table list parameters for nudr-perf-info microservice.

Table 3-11 nudr-perf-info microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
replicaCount	Indicates the number pods which need to be created as part of deployment.	2	Valid number
image	Name of the docker image of perf info service	perf-info	NA
imageTag	Tag of the image of perf info service	24.3.0	NA
imagePullPolicy	Indicates if the image needs to be pulled or not	Always	Possible Values - Always IfNotPresent Never
dbHookImage.name	Image name of dbHook	common_config_hook	Not Applicable
dbHookImage.tag	Image tag name of dbHook	24.3.0	Not Applicable

Table 3-11 (Cont.) nudr-perf-info microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
dbHookImage.pullPolicy	Pull policy of image	IfNotPresent	Possible Values - Always IfNotPresent Never
service.type	K8s service type	ClusterIP	Possible Values- ClusterIP NodePort LoadBalancer
global.logStorage	Log storage for ephemeral storage request	*containersLogStorageRequestsRef	Unit MB Do not change reference unless a different value needs to be configured
global.crictlStorage	Crictl storage for ephemeral storage request	*containersCrictlStorageRequestsRef	Unit MB Do not change reference unless a different value needs to be configured
resources.limits.cpu	CPU Limit	1	NA
resources.limits.memory	Memory Limit	1Gi	NA
resources.limits.ephemeralStorage	Ephemeral Storage allocation limits	1024Mi	Unit MB
resources.requests.cpu	CPU Requested	1	NA
resources.requests.memory	Memory Requested	1Gi	NA
nodeSelector	Node selector to stick to one slave node	{}	NA
tolerations	Pod Toleration	[]	NA
affinity	Pod Affinity configurations	{}	NA
commonCfgClient.enabled	Set it to true if persistent configuration needs to be enabled.	true	true/false
commonCfgServer.configServerSvcName	Service name of common configuration service to which the client tries to poll for configuration updates	nudr-config	Valid server service name
commonCfgServer.host	Host name of Common configuration server to which client tries to poll for configuration updates. This value is picked up if commonCfgServer.configServerSvcName is not available	10.75.224.123	Valid host details
commonCfgServer.port	Port of Common Configuration server	5001	Valid Port

Table 3-11 (Cont.) nudr-perf-info microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
commonCfgServer.pollingInterval	This is the interval between two subsequent polling requests from config client to server	5000	Valid period
commonServiceName	This is the common service name that is currently requesting for configuration updates from server	alt-route	Not Applicable
ingress.enabled	Ingress flag control	false	true/false
configmapPerformance.prometheus	Prometheus server k8s service URL Information	http://occne-prometheus-server.occne-infra:80	Not Applicable
configmapPerformance.jaeger	Jaeger Agent K8s service URL Information	occne-tracer-jaeger-agent.occne-infra	Not Applicable
configmapPerformance.jaeger_query_url	Jaeger Query k8s service URL Information	http://occne-tracer-jaeger-query.occne-infra	Not Applicable
overloadManager.perfInfoScrapeInterval	Overlaod Manager Scrape Interval	30	Unit: Seconds
log.level.perfinfo	Log level for perf-info service	WARN	Possible Values - WARN INFO DEBUG
istioSidecarQuitUrl	Quit url configurable for side car, referred to global section istioSidecarQuitUrl. Change only if the url is different for the side car container in this microservice	*istioQuitUrl	Not ApplicableNote: Do not change this value unless it is different from the value configured in global section reference
istioSidecarReadyUrl	Readiness url configurable for side car, referred to global section istioSidecarReadyUrl. Change only if the url is different for the side car container in this microservice	*istioReadyUrl	Not ApplicableNote: Do not change this value unless it is different from the value configured in global section reference
serviceMeshCheck	Enable when deployed in serviceMesh, referred to the serviceMeshCheck in global section. Set to false when side car is not included for this service	*serviceMeshFlag	Not ApplicableNote: Do not change this value unless it is different from the value configured in global section reference
overloadManager.enabled	Enable when overload manager is enabled.	*overloadEnabled	Not Applicable. Do not change this value unless it is different from the value configured in global section.

Table 3-11 (Cont.) nudr-perf-info microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
overloadManager.adjacentLevelDuration	Based on this, overload threshold level happens.	40	Not Changed
overloadManager.ingressGatewaySvcName	Default svcName for Ingress Gateway	ingressgateway	-
overloadManager.ingressGatewayHost	This parameter can be left blank as svcName is configured.	-	Not Changed
overloadManager.ingressGatewayPort	Default svcName for Ingress Gateway.	80	Default port for ingressgateway.
overloadManager.ingressGatewayScrapeInterval	Based on this, overload manager send requests to Ingress Gateway. By default, it is 1 second.	1	It can be changed to different value.
overloadManager.ingressGatewayFailureRateLength	Based on this, failure rate time series is formed.	60	It can be changed to different value.
overloadManager.perfInfoScrapeInterval	Based on this parameter, perfinfo gets CPU and memory from perf-info itself.	10	It can be changed to different value.
overloadManager.nfType	NF Type value should be UDR	UDR	This value should not be changed.
tagNamespace	Dimension names used for metrics.	namespace	Values for CNE 1.8 or OSO are: {tagNamespace: kubernetes_namespace} Values for CNE 1.9 and above is: {tagNamespace: namespace}
tagContainerName	Dimension names used for metrics.	container	Values for CNE 1.8 or OSO are: {tagContainerName: container_name}Value for CNE 1.9 and above is: {tagContainerName: container}
tagServiceName	Dimension names used for metrics.	service	Values for CNE 1.8 or OSO are: {tagServiceName: kubernetes_name}Value for CNE 1.9 and above is: {tagServiceName: service}
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED

Table 3-11 (Cont.) nudr-perf-info microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level . If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED

tolerations	When tolerationsSetting is ENABLED, configure tolerations here.	[]	Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect:`
			`"NoSchedule"`

helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2
nodeSelectorEnabled	Flag to enable v1 nodeSelector	false	true/false
nodeSelectorKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	''	Valid key for a label for a particular node
nodeSelectorValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	''	Valid value pair for the above key for a label for a particular node

Table 3-11 (Cont.) nudr-perf-info microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2. Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod
envMysqlDatabase	Database name in which the leader election table needs to be created	`udrConfigDB`	NA
envLeaderElectionTableName	Table name to be used for the leader election table	`leader_election`	NA
maxUnavailable	Number of replicas that can go down during a disruption	50%	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	30s	Unit: Seconds

3.1.11 diam-gateway Parameters

Following table list configurable parameters for diam-gateway microservice.

Table 3-12 diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
global.servicePorts.diamGatewayHttp	HTTP Service Port exposed	8000	Valid Port	NA
global.servicePorts.diamGatewayDiameter	Diameter Service Port	3868	Valid Port	NA
global.servicePorts.configServerHttp	Config server connection Service port	5807	Valid Port	NA
global.servicePorts.bindingHttp	Binding Service Port	8080	Valid Port	NA
global.servicePorts.diamGatewayDistCache	Diameter Cache Service port	5801	Valid Port	NA
global.servicePorts.dbConnStatusHttp	Database connection check Service port	8000	Valid Port	NA
global.containerPorts.monitoringHttp	Monitoring Container Port	9000	Valid Port	NA

Table 3-12 (Cont.) diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
global.containerPorts.diamGatewayHttp	HTTP Container Port	8000	Valid Port	NA
global.containerPorts.diamGatewayDiameter	Diameter Connection Container Port	3868	Valid Port	NA
global.containerPorts.diamGatewayDistCache	Diameter Cache Container Port	5801	Valid Port	NA
global.containerPorts.configContainerSignalingHttp	Signaling Container Port	8100	Valid Port	NA
global.containerPorts.configContainerMonitoringHttp	Configuration Monitoring Container Port	8101	Valid Port	NA
image	Docker image name	diam-gateway	not applicable	NA
imageTag	Docker image tag	24.3.0	not applicable	not applicable
replicas	replicas of diam-gateway	2	not applicable	not applicable
maxUnavailable	Number of replicas that can go down during a disruption	1	not applicable	This parameter is used for PodDisruptionBudget Kubernetes resource
envLoggingLevelApp	diam-gateway log level	INFO	DEBUG, INFO, WARN, ERROR	not applicable
envDiameterPort	diam-gateway diameter service port	3868	valid port number	not applicable
envDiameterRealm	diameter server realm	*diamGatewayRealm	not applicable	not applicable
envDiameterIdentity	diameter server identity	*diamGatewayIdentity	not applicable	not applicable
envDiameterIOThreadCount	Number of thread for IO operation	12	0 to 2* CPU	not applicable
envDiameterIOQueueSize	Queue size for IO	8192	2048 to 8192	not applicable
envDiameterMsgBufferThreadCount	Number of threads for process the message	24	0 to 2* CPU	not applicable
envDiameterMsgBufferQueueSize	Queue size of processing message	4096	1024 to 4096	not applicable
envDiameterValidationStrictParsing	Strict parsing of diameter AVP and messages	false	not applicable	strict parsing

Table 3-12 (Cont.) diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
envDiameterHostIp	Host IP value in capability exchange message	""	not applicable	If the server wants to validate host IP address and need to add all Kubernetes node name and IP address. format: nodename=ipaddress
enableConfigService	config server support for diam-gateway	*configServerEnabled	true/false	not applicable
resources.limits.cpu	max CPU allotment for diam-gateway pod	2	not applicable	not applicable
resources.limits.memory	Maximum memory allotment for diam-gateway pod	2	not applicable	not applicable
resources.limits.ephemeral-storage	Ephemeral storage allocation limits	1024Mi	Unit: MB	not applicable
resources.requests.cpu	CPU allocated for diam-gateway pod during deployment	2	not applicable	not applicable
resources.requests.memory	Memory allocated for diam-gateway pod during deployment	2	not applicable	not applicable
resources.requests.ephemeral-storage	Ephemeral storage allocation request	72Mi	Unit MB	not applicable
service.type	Kubernetes diam-gateway service type for exposing UDR deployment	LoadBalancer	ClusterIP NodePort LoadBalancer	Recommends to set LoadBalancer as default value always
service.customExtension.labels	Custom labels that needs to be added to nudr-diam-gateway specific service.	null	NA	This can be used to add custom label(s) to nudr-diam-gateway deployment.
service.customExtension.annotations	Custom annotations that needs to be added to nudr-diam-gateway specific service.	null	NA	This can be used to add custom annotation(s) to nudr-diam-gateway deployment.

Table 3-12 (Cont.) diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
service.externalTrafficPolicy	Restricts the LBVM to use only the IP of local worker node for routing traffic on which diameter gateway is deployed.	Cluster	Cluster/Local	
deployment.customExtension.labels	Custom labels that needs to be added to nudr-diam-gateway specific deployment.	null	NA	This can be used to add custom label(s) to nudr-diam-gateway deployment.
deployment.customExtension.annotations	Custom annotations that needs to be added to nudr-diam-gateway specific deployment.	null	NA	This can be used to add custom annotation(s) to nudr-diam-gateway deployment.
fullnameOverride	pod-name in deployment	diam-gateway	not applicable	not applicable
peerConfig.setting	Diameter peer setting	reconnectDelay: 3responseTimeout: 4connectionTimeOut: 3watchdogInterval: 6transport: 'TCP'reconnectLimit: 50	Not Applicable	1. Reconnect delay for diameter reconnect (in seconds). 2. total turnaround time for process the diameter messages (in sec). 3. TCP connection timeout time (in sec). 4. DWR and DWA messages every number of time (in sec) 5. Transport layer 6. reconnect the number of time if diameter peer is down

Table 3-12 (Cont.) diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
readinessProbe.initialDelaySeconds	Configurable wait time before performing the first readiness probe by the kubelet	80	Not Applicable Unit: Seconds	Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters.
readinessProbe.periodSeconds	Time interval for every readiness probe check.	5	Not Applicable Unit: Seconds	Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters.
resources.requests.ephemeral-storage	Ephemeral storage allocation request.	72Mi	Unit: MB	Not Applicable
livenessProbe.initialDelaySeconds	Configurable wait time before performing the first liveness probe by the kubelet.	80	Not Applicable Unit: Seconds	Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters.
livenessProbe.periodSeconds	Time interval for every liveness probe check.	20	Not Applicable Unit: Seconds	Do not change this value. If there is delay in pod coming up and probe is killing the pod then tune these parameters.
internalTracingEnabled	Flag to enable/ disable jaeger tracing for nudr-ondemand-migration-service	false	true/false	NA
podCongetion.enabled	diameter-gateway pod congestion feature enable flag	false	true/false	NA
notificationRetryCode.diameter	PNR retry error code configurations	3002,3004,3005,3006,4003,5012	Valid diameter error code	
notificationRetryCode.http	PNR retry error code configurations	*notificationRetryErrorCode	NA	Do not change reference

Table 3-12 (Cont.) diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
peerConfig.clientPeers	Diameter client node information	`- realm: 'seagull1.com' identity: 'seagull1a.seagull1.com' - realm: 'west.com' identity: 'seagull.west.com'`	Information should be yaml list	NA
peerConfig.serverPeers	Diameter server peer node information	null	Information should be yaml list	NA
extraContainers	This configuration decides at this microservice level the extra container need to be used or not.	USE_GLOBAL_VALUE	Allowed values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	This configuration decides service level extraContainer support. The default value lets it to be dependent on what is configured in the global level
serviceMeshCheck	Enable when deployed in serviceMesh, referred to the serviceMeshCheck in global section. Set to false when side car is not included for this service	*serviceMeshFlag	Not Applicable	Do not change this value unless it is different from the value configured in global section reference
istioSidecarReadyUrl	Readiness url configurable for side car, referred to global section istioSidecarReadyUrl. Change only if the URL is different for the side car container in this microservice	*istioReadyUrl	Not Applicable	Do not change this value unless it is different from the value configured in global section reference

Table 3-12 (Cont.) diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
podCongestion.enabled	Enable or disable the Diameter Gateway Pod Congestion feature	false	true/false	NA
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

tolerations	When tolerationsSetting is ENABLED, configure tolerations here.	[]	Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

Table 3-12 (Cont.) diam-gateway microservice parameters

Parameter	Description	Default Value	Range or Possible Values (If applicable)	Notes
helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting.	v1	Allowed Values: • v1 • v2	NA
nodeSelector.nodeKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid key for a label for a particular node.	NA
nodeSelector.nodeValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid value pair for the above key for a label for a particular node.	NA
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2. Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod	NA
gracefulShutdown.gracePeriod	The graceful shutdown period used for Diameter Gateway service termination.	30s	Unit Seconds	NA

3.1.12 ocudr-ingressgateway-sig Microservice Parameters

The following table provides parameters for ocudr-ingressgateway microservice (API Gateway)

Table 3-13 ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.type	ocudr-ingressgateway service type	LoadBalancer	Possible Values: ClusterIP NodePort LoadBalancer	NA
global.metalLbIpAllocationEnabled	Flag to enable Address Pool for Metallb	true	true/false	NA
global.metalLbIpAllocationAnnotation	Address Pool for Metallb	metallb.universe.tf/address-pool: signaling	NA	NA
global.staticNodePortEnabled	Flag to enable static node port	false	NA	NA
global.istioIngressTlsSupport.ingressGateway	Flag to enable clear text traffic from outside the cluster when Service Mesh is enabled	false	true/false	NA
global.xfccHeaderValidation.validation.enabled	Flag to determine if incoming xfcc header needs to be validated	&xfccValidationEnabled false	true/false	NA
global.xfccHeaderValidation.validation.nfList	List of configured NF FQDN's against which the matchField parameter present in the XFCC header of incoming request is validated	nfList: -scp1.com -scp2.com	Valid Network Function List, which is available as part of XFCC Header	NA
global.xfccHeaderValidation.validation.matchCerts	The number of certificates that need to be validated starting from the right most entry in the XFCC header	-1	(Value = -1 validation to be performed against all entries, value = +ve number, then validate starting from the right most entry backwards)	NA
global.xfccHeaderValidation.validation.matchField	Field in XFCC header against which the configured nfList FQDN validation needs to be performed	Subject.CN	Valid XFCC header field	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.xfccHeaderValidation.validation.errorCodeOnValidationFailure	Field determines the errorCode to be sent in response when XFCC header validation fails at Ingress Gateway	500	Valid HTTP error code	NA
global.publicHttpsSignalingPort	Https Signaling port	Yes	443	NA
openTracing.jaeger.enableb3Propagation	To send b3 zipkin headers instead of uber-trace-id	Yes (If jaegerTracingEnabled is true)	false	NA
dbConfig.backupDbName	configure when your backup table should have separate schema	No	NA	NA
isIpv6Enabled	Set this flag to true when deployed in Ipv6 enabled setup	No	NA	NA
pod-protection.enabled	podProtection feature can be enabled and disabled using this flag	Yes	false	NA
pod-protection.monitoringInterval	Interval in which the state of the resource is checked and updated accordingly	Yes	100 (ms)	NA
pod-protection.congestionControl.enabled	the congestion control mechanism of pod protection can be enabled and disabled using this flag	Yes	false	NA
pod-protection.congestionControl.stateChangeSampleCount	the number of times the pod has to remain in same state in order to transition to different state	Yes	10	NA
pod-protection.congestionControl.actionSamplingPeriod	actionSamplingPeriod the (isSchedulerTimeSensitiveAction) action will be executed only on the specified time (actionSamplingPeriod * monitoringInterval)	Yes	3	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
pod-protection.congestionControl.states[i].name	Any meaning full name of the congestion state	Yes	Normal at index 0, Doc at index 2, Congested at index 3	NA
pod-protection.congestionControl.states[i].weight	Each state is identified by the weight. The weight defines the criticality of the state. Lower the state lower the criticality	Yes	0, at index 0, 1 at index 1, 2 at index 2	NA
pod-protection.congestionControl.states[i].entryAction[j].action	The action that has to be executed at the current state	Yes	MaxConcurrentStreamsUpdate, AcceptIncomingConnections	NA
pod-protection.congestionControl.states[i].entryAction[j].args	the argument corresponding to action	Yes	NA	NA
global.xfccHeaderValidation.validation.peerList	Supports virtual host and static FQDNS both. It is backward compatible with nfLIst(old configuration).		NA	NA
global.xfccHeaderValidation.validation.errorDescriptionOnValidationFailure	Configurable error description to be returned when XFCC header validation fails at Ingress Gateway. This error is populated in ProblemDetails response in **ProblemDetails.detail** section	Validation of SCP failed	NA	NA
global.xfccHeaderValidation.validation.errorCauseOnValidationFailure	Configurable error cause to be returned when XFCC header validation fails at Ingress Gateway. This error is populated in ProblemDetails response in **ProblemDetails.cause** section	XFCC Validation Failed	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.xfccHeaderValidation.validation.errorTitleOnValidationFailure	Configurable error title to be returned when XFCC header validation fails at Ingress Gateway. This error is populated in ProblemDetails response in **ProblemDetails.title** section.	Internal Server Error	NA	NA
global.xfccHeaderValidation.validation.retryAfter	Retry-after value in seconds populated in Retry-After header and sent in response, when XFCC header validation fails at Ingress Gateway and the value configured for errorCodeOnValidationFaulire lies in 3xx series	""	Retry-After value in seconds	NA
global.xfccHeaderValidation.validation.redirectUrl	Redirect-Url string populated in LOCATION header and sent in response when XFCC header validation fails at Ingress Gateway and the value configured for errorCodeOnValidationFaulire lies in 3xx series	""	Valid redirect url	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

xfcc.xfccHeaderValidation.validation.errorTrigger	Configurable exception or error type and details for an error scenario in Ingress Gateway for XFCC	errorTrigger:
		  -
		exceptionType:
		XFCC_HEADER_NOT_PRESENT_OR_EMPTY
		errorCode:
		"500"
		errorDescription:
		"Validation of SCP failed"
		errorCause:
		xfcc header is not present or empty in the request
		errorTitle:
		"Internal Server Error"
		retryAfter:
		""
		redirectUrl:
		""
		  -
		exceptionType:
		XFCC_MATCHCERTCOUNT_GREATER_THAN_CERTS_IN_HEADER
		errorCode:
		"500"
		errorDescription:
			NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		"Validation of SCP failed" errorCause: matchCerts count is greater than the certs in the request errorTitle: "Internal Server Error" retryAfter: "" redirectUrl: "" - exceptionType: XFCC_HEADER_INVALID errorCode: "500" errorDescription: "Validation of SCP failed" errorCause: xfcc header is invalid errorTitle: "Internal Server Error" retryAfter: ""		

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		`redirectUrl: ""`		
image.name	Docker image name	ocingress_gateway	NA	NA
image.tag	Tag or version of Image	24.3.3	NA	NA
image.pullPolicy	This setting indicates if image needs to be pulled or not	IfNotPresent	Possible Values: Always IfNotPresent Never	NA
initContainersImage.name	Docker Image name	configurationinit	NA	NA
initContainersImage.tag	Tag or version of Image	24.3.3	NA	NA
initContainersImage.pullPolicy	This setting indicates if image need to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
updateContainersImage.name	Docker Image name	configurationupdate	NA	NA
updateContainersImage.tag	Tag or version of Image	24.3.3	NA	NA
updateContainersImage.pullPolicy	This setting indicates if image need to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
dbHookImage.name	Image name of dbHook	common_config_hook	NA	NA
dbHookImage.tag	Image tag name of dbHook	24.3.3	NA	NA
dbHookImage.pullPolicy	Indicates the pull policy of image	IfNotPresent	Possible Values: Always IfNotPresent Never	NA
service.ssl.tlsVersion	TLS version to be used	TLSv1.2, TLSv1.3	Valid TLS version	These are service fixed parameters
service.ssl.privateKey.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.privateKey.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.ssl.privateKey.rsa.fileName	RSA private key stored in the secret	rsa_private_key_pkcs1.pem	NA	NA
service.ssl.privateKey.ecdsa.fileName	ECDSA private key stored in the secret	ecdsa_private_key_pkcs8.pem	NA	NA
service.ssl.certificate.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.certificate.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.certificate.rsa.fileName	RSA certificate stored in the secret	apigatewayrsa.cer	NA	NA
service.ssl.certificate.ecdsa.fileName	ECDSA certificate stored in the secret	apigatewayecdsa.cer	NA	NA
service.ssl.caBundle.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.caBundle.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.caBundle.fileName	Name of caBundle file stored in the secret	caroot.cer	NA	NA
service.ssl.keyStorePassword.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.keyStorePassword.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.keyStorePassword.fileName	Keystore password stored in the secret	key.txt	NA	NA
service.ssl.trustStorePassword.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.trustStorePassword.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.trustStorePassword.fileName	trustStore password stored in the secret	trust.txt	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.initialAlgorithm	Algorithm to be used for SSL verification when request lands on secure port of Ingress Gateway. You can use ES256, but use its corresponding certificates also.	RS256	RS256/ES256	NA
http1.client.keepAlive	http1.1 settings when enablehttp1 is set to true. If keepAlive is set as false, then the connection is closed after one request.	false	true/false	NA
http1.client.useConnectionPool	http1.1 settings when enablehttp1 is set to true. If useConnectionPool is set as false, it creates new connection instead of looking for active connections	false	true/false	NA
resources.limits.cpu	CPU allotment limitation	2	NA	NA
resources.limits.memory	Memory allotment limitation	2Gi	NA	NA
resources.limits.initServiceCpu	Maximum number of CPUs that Kubernetes allow the ingress-gateway init container to use	1	NA	NA
resources.limits.initServiceMemory	Memory limit for ingress-gateway init container	1Gi	NA	NA
resources.limits.updateServiceCpu	Maximum number of CPUs that Kubernetes allow the ingress-gateway update container to use	1	NA	NA
resources.limits.updateServiceMemory	Memory limit for ingress-gateway update container	1Gi	NA	NA
resources.limits.commonHooksCpu	CPU Limit of database Hook Container	1	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.limits.commonHooksMemory	Memory limit of database Hook Container	1Gi	NA	NA
resources.requests.cpu	Number of CPUs allocated for ocudr-endpoint pod	2	NA	NA
resources.requests.memory	Memory allocated for ocudr-endpoint pod	2Gi	NA	NA
resources.requests.initServiceCpu	Number of CPUs that the system guarantees for ingress-gateway init container, and Kubernetes uses this value to decide the node to place the pod	1	NA	NA
resources.requests.initServiceMemory	Memory allotment that the system guarantees for ingress-gateway init container, and Kubernetes uses this value to decide the node to place the pod	1Gi	NA	NA
resources.requests.updateServiceCpu	Number of CPUs that the system guarantees for ingress-gateway update container, and Kubernetes uses this value to decide the node to place the pod	1	NA	NA
resources.requests.updateServiceMemory	Memory allotment that the system guarantees for ingress-gateway update container, and Kubernetes uses this value to decide the node to place the pod	1Gi	NA	NA
resources.requests.commonHooksCpu	Db Hook Container CPU request, **and K8s** uses this value to decide on which node to place the pod.	1	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.requests.commonHooksMemory	Db Hook Container CPU request, **and K8s** uses this value to decide on which node to place the pod.	1Gi	NA	NA
resources.target.averageCpuUtil	CPU utilization limit for auto scaling	80	NA	NA
maxUnavailable	Number of pods always running	1	NA	NA
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	2	NA	NA
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	5	NA	NA
log.level.root	Log level shown on ocudr-endpoint pod	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.ingress	Log level shown on ocudr-ingressgateway pod for ingress related flows	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.oauth	Log level shown on ocudr-ingressgateway pod for oauth related flows	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.updateContainer	Log level for Update container	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.configclient	Log level for Config Client	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.hook	Log level for hook	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.cncc.security	Log level for cncc logs	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.traceIdGenerationEnabled	Flag to enable TraceId Generation	true	true/false	NA
initssl	Flag to initialize SSL related infrastructure in init/update container	false	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
jaegerTracingEnabled	Flag to enable Jaeger Tracing	false	true/false	NA
openTracing.jaeger.udpSender.host	FQDN of Jaeger agent service	occne-tracer-jaeger-agent.occne-infra	Valid FQDN	NA
openTracing.jaeger.udpSender.port	UDP port of Jaeger agent service	6831	Valid Port	NA
openTracing.jaeger.probabilisticSampler	Sampler makes a random sampling decision with the probability of sampling. This parameter indicates the probabilistic Sampler on Jaeger.	0.5	Range: 0.0 - 1.0	For example, if the value set is 0.1, approximately 1 in 10 traces are sampled.
ciphersuites	Supported cipher suites for SSL	- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	NA	NA
oauthValidatorEnabled	Flag to enable OAUTH validator	false	NA	NA
nfType	NF type of the service producer	UDR	NA	Mandatory when oauthValidatorEnabled is true
producerScope	Comma separated list of services hosted by service producer	nudr-dr,nudr-group-id-map	Valid service list	Mandatory when oauthValidatorEnabled is true

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
allowedClockSkewSeconds	Set this value if clock on the parsing NF (producer) is not perfectly in sync with the clock on the consumer NF that created the JWT	0 Unit: Seconds	NA	Mandatory when oauthValidatorEnabled is true
enableInstanceIdConfigHook	Flag to enable a preUpgrade hook when persistent config is enabled. It populates InstanceId configuration based on the content of nrfPublicKeyKubeSecret if instance id configuration existed for previous release then that is picked	true	true/false	NA
nrfPublicKeyKubeSecret	Name of the secret that stores public key(s) of NRF	oauthsecret	NA	Mandatory when oauthValidatorEnabled is true
nrfPublicKeyKubeNamespace	Namespace of the NRF publicKey secret	ocudr	NA	Mandatory when oauthValidatorEnabled is true
validationType	It can be either "strict" or "relaxed". Strict means that incoming requests without "Authorization"(Access Token) header are rejected. Relaxed means that if incoming request contains "Authorization" header, it is validated. If incoming request does not contain "Authorization" header, validation is ignored	strict	strict/relaxed	Mandatory when oauthValidatorEnabled is true
producerPlmnMNC	MNC of service producer	14	Valid MNC	NA
producerPlmnMCC	MCC of service producer	310	Valid MCC	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
errorCodeOnValidationFailure	Error code to return on oauth validation failure	401	Valid http error code	NA
errorCodeOnTokenAbsence	Error code to return when oauth token is not present	400	Valid http error code	NA
oauthErrorConfig.errorTitle	Error title information to return in error response Problem details	InvalidOAuthAccessToken	NA	NA
oauthErrorConfig.errorDescription	Error description to return in error response Problem details	UNAUTHORIZED	NA	NA
oauthErrorConfig.errorCause	Error cause to return in response	""	NA	NA
oauthErrorConfig.redirectUrl	Redirect URL configuration	null	NA	NA
oauthErrorConfig.retryAfter	Value of retryAfter to be configured	null	NA	NA
enableIncomingHttp	Flag to enable HTTP requests	true	NA	NA
enableIncomingHttps	Flag to enable HTTPS requests	false	true or false	NA
enableOutgoingHttps	Flag to enable outgoing HTTPS requests	false	true or false	NA
maxConcurrentPushedStreams	Jetty client settings for maximum concurrent streams	6000	Valid Number	NA
maxConnectionsPerDestination	Jetty client settings for maximum connections per destination	10	Valid Number	NA
maxRequestsQueuedPerDestination	Queue Size at the ocudr-endpoint pod	5000	Valid Number	NA
maxConnectionsPerIp	Number of connections from endpoint to other microservices	10	Valid Number	NA
connectionTimeout	Connection timeout limit configured on ingress gateway client towards the backend	10000 Unit: Milliseconds	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
requestTimeout	Request timeout limit configured on ingress gateway client towards the backend	1000 Unit: Milliseconds	NA	NA
serviceMeshCheck	Flag to check service mesh	*serviceMeshFlag	true/false	NA
istioSidecarQuitUrl	Quit URL configured for side car	*istioQuitUrl	Valid url	NA
istioSidecarReadyUrl	Readiness URL configured for side car	*istioReadyUrl	Valid url **Note:** The port used should be the admin configured for istio proxy container.	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
routesConfig	Routes configured to connect to different microservices of UDR	`# configuring routes routesConfig: - id: traffic_mapping_http uri: http://{{ .Release.Name }}-nudr-drservice:5001 path: /nudr-dr/** order: 1 metadata: # oauthValidation route level oauthValidator: enabled: *oauthEnabled # xfccHeaderValidation route level xfccHeaderValidation: # When enabled on global section, the same gets referred here # Set to false, if needs to be disabled for this particular route`	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		validationEnabled: *xfccValidationEnabled - id: traffic_mapping_http_udsf uri: http://{{ .Release.Name }}-nudr-drservice:5001 path: /nudsf-dr/** order: 2 metadata: # oauthValidation route level oauthValidator: enabled: *oauthEnabled # xfccHeaderValidation route level xfccHeaderValidation: # When enabled on global section, the same gets referred here # Set to false, if needs to be disabled for this particular route		

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		validationEnabled: *xfccValidationEnabled - id: traffic_mapping_http_group uri: http://{{ .Release.Name }}-nudr-drservice:5001 path: /nudr-group-id-map/** order: 3 metadata: # oauthValidation route level oauthValidator: enabled: *oauthEnabled # xfccHeaderValidation route level xfccHeaderValidation: # When enabled on global section, the same gets referred here # Set to false, if needs to be disabled for this particular		

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

		route
		
		validationEnabled: *xfccValidationEnabled
		- id: traffic_mapping_http_eir
		  uri: http://{{ .Release.Name }}-nudr-drservice:5001
		  path: /n5g-eir-eic/**
		  order: 4
		  metadata:
		  # oauthValidation route level
		
		oauthValidator:
		
		enabled: *oauthEnabled
		  # xfccHeaderValidation route level
		
		xfccHeaderValidation:
		  # When enabled on global section, the same gets referred here
		  # Set to false, if needs to be disabled for this particular		

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

		`route`
		
		`validationEnabled: *xfccValidationEnabled`		

service.customExtension.labels	Use this to add custom label(s) to ingressgateway service	null	NA	NA
service.customExtension.annotations	Use this to add custom annotation(s) to ingress gateway service	null	NA	NA
deployment.customExtension.labels	Use to add custom label(s) to ingress gateway deployment	null	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	This can be used to add custom annotation(s) to ingressgateway-sig specific deployment	`annotations:`
		
		`sidecar.istio.io/proxyCPU: "3000m"`
		
		`sidecar.istio.io/proxyCPULimit: "3000m"`
		
		`sidecar.istio.io/proxyMemory: "1Gi"`
		
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		
		`proxy.istio.io/config: |`
		
		`terminationDrainDuration: 60s`
			NA	NA

readinessProbe.initialDelaySeconds	Configurable wait time before performing the first readiness probe by Kubelet	30 Unit: Seconds	NA	Do not change this value. If you see delays in pod to come up and probe is killing the pod then you should tune these parameters
readinessProbe.periodSeconds	Time interval for every readiness probe check	10 Unit: Seconds	NA	Do not change this value. If you see delays in pod to come up and probe is killing the pod then you should tune these parameters
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	NA	Do not change this default value

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	1	NA	Do not change this default value
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes indicates failureThreshold time before giving up	3	NA	Do not change this default value
livenessProbe.initialDelaySeconds	Configurable wait time before performing first liveness probe by Kubelet	30 Unit: Seconds	NA	Do not change this value. If you see delays in pod to come up and probe is killing the pod then you should tune these parameters
livenessProbe.periodSeconds	Time interval for every liveness probe check	15 Unit: Seconds	NA	Do not change this value. If you see delays in pod to come up and probe is killing the pod then you should tune these parameters
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	NA	Do not change this value
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	1	NA	Do not change this value
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes tries failureThreshold times before giving up	3	NA	Do not change this value

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainers	This configuration decides service level extraContainer support. The default value is dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA
commonCfgClient.enabled	Flag to enable persistent configuration	true	true/false	NA
commonCfgServer.configServerSvcName	Service name of common configuration service to which the client tries to poll for configuration updates	nudr-config	Valid server service name	NA
commonCfgServer.host	Host name of common configuration server to which client tries to poll for configuration updates	10.75.224.123	Valid host details	This value is picked up if commonCfgServer.configServerSvcName is not available
commonCfgServer.port	Port of Common Configuration server	5001	Valid port	NA
commonCfgServer.pollingInterval	Interval between two subsequent polling requests from configuration client to server	5000	Valid period	NA
commonServiceName	Common service name that is currently requesting for configuration updates from server	igw	NA	NA
userAgentHeaderValidationConfigMode	This flag is used to accept the user-agent configurations from HELM or REST	HELM	HELM/REST	NA
userAgentHeaderValidation.enabled	This flag is used to enable the user-agent feature	false	true/false	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
userAgentHeaderValidation.consumerNfTypes	This parameter is used to configure the NF Types, like PCF, NRF, UDM that are using UDR services	EMPTY	All NF Types that are defined in 3gpp specification. Example PCF,NRF,NEF,UDM	NA
restoreBackupOnInstall	Flag to enable restore when data is picked up from backup table during installation of ingress gateway	false	true/false	NA
dropTablesOnUpgrade	This flag when enabled drops the common configuration tables on Helm upgrade if there is no data in the tables	false	true/false	NA
autoRedirect	Flag to enable redirection in Ingress Gateway. If enabled, Ingress Gateway redirects to URL present in location header when the response code is 301, 302, 303, 307, or 308	true	true/false	NA
serverHeaderConfigMode.serverHeaderDetails.enabled	This flag is to enable or disable server header.	false	true/false	NA
global.logStorage	Ephemeral storage configuration for log storage	No	0	NA
global.crictlStorage	Ephemeral storage configuration for crictl storage	No	0	NA
podProtectionConfigMode	This configuration is to enable pod protection configuration to rest based or helm based.	NA	HELM or REST	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

nfSpecificConfig	This configuration is to enable or disable the NF specific default value configuration and a list of configuration files is defined in the default configuration list.	NA	nfSpecificConfig:
			  enabled: false
			
			featureList:
			  - podprotection
			  - trafficrejectmode
			  - readinessconfig
			  - errorcodeserieslist
			  - messagelogging
			  - serverheaderdetails
			  - routelevelratelimiting
			  - ocpolicymapping
			  - routesconfiguration
			  - ocdiscardpolicies
			  - errorcodeprofiles
			  - oauthvalidatorconfiguration
			  - logging
				NA

ephemeralStorageLimit	Ephemeral storage limit	No	0	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.publicHttp1SignalingPort	HTTP1 Signaling port	-	NA	NA
global.staticHttp1NodePort	HTTP1 Node Port	-	NA	NA
global.containerHttp1Port	Container Http1 port	-	NA	NA
global.enableIncomingHttp1	global flag to enable or disable HTTP1	-	NA	NA
global.enableTLSIncomingHttp1	global flag to enable or disable HTTP1 in TLS or NON TLS mode	-	NA	NA
userAgentHeaderValidationConfigMode	parameter to configure mode of configuration. It can be either REST or HELM	YES	Helm	NA
userAgentHeaderValidation.enabled	flag to enable or disable the feature	NO	False	NA
userAgentHeaderValidation.validationType	parameter to decide the type of validation when User-Agent header is not present in the incoming request	NO	NA	NA
userAgentHeaderValidation.consumerNfTypes	list of valid consumer NF types as per 3GPP specification	NO	NA	NA
global.lciHeaderConfig.enabled	Flag to enable or disable the LCI headers reporting	false	true or false	NA
global.lciHeaderConfig.loadThreshold	If current load level is beyond the previously computed load level + loadThreshold, LCI headers are reported again.	40	NA	NA
global.lciHeaderConfig.localLciHeaderValidity	Validity period of the LCI headers reported to the consumer NF. LCI headers are reported again if the headers are reported as previously expired.	1000 (value in milliseconds)	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.lciHeaderConfig.producerSvcIdHeader	Header name, which holds the producer svc identity.	`custom_producer_svc_header`	NA	NA
global.ociHeaderConfig.enabled	Flag to enable or disable the OCI headers reporting	false	true or false	NA
global.ociHeaderConfig.producerSvcIdHeader	Header name, which holds the producer svc identity.	custom_producer_svc_header	NA	NA
global.ociHeaderConfig.validityPeriod	Validity period of the OCI headers reported to the consumer NF. OCI headers are reported again if the headers are reported as previously expired.	5000 (value in milliseconds)	NA	NA
global.ociHeaderConfig.overloadConfigRange.minor	Range to identify the minor overload condition	"[0-10]"	NA	NA
global.ociHeaderConfig.overloadConfigRange.major	Range to identify the major overload condition	"[10-70]"	NA	NA
global.ociHeaderConfig.overloadConfigRange.critical	Range to identify the critical overload condition	"[70-100]"	NA	NA
global.ociHeaderConfig.reductionMetrics.minor	Reduction metric to be reported for the minor overload condition	5 (Possible values 1 to 9 both inclusive)	NA	NA
global.ociHeaderConfig.reductionMetrics.major	Reduction metric to be reported for the major overload condition	10 (Possible values 5 to 15 both inclusive)	NA	NA
global.ociHeaderConfig.reductionMetrics.critical	Reduction metric to be reported for the critical overload condition	30 (Possible values 10 to 50 both inclusive)	NA	NA
global.nfInstanceId	NF instance Id of the producer NF	6faf1bbc-6e4a-4454-a507-a14ef8e1bc11	NA	NA

Table 3-13 (Cont.) ocudr-ingressgateway microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.svcToSvcInstanceIdMapping.svcName	Back-end service name, which should match the producerSvcIdHeader value and perf info reported service name for the LCI or OCI headers reporting.	nf-registration	NA	NA
global.svcToSvcInstanceIdMapping.serviceInstanceId	Back-end service instance id to be included in the LCI or OCI headers	fe7d992b-0541-4c7d-ab84-c6d70b1b01b1	NA	NA
global.perfInfoConfig.pollingInterval	Configurable interval at which the load infromation is polled from the perf-info service at GW	5000	NA	NA
global.perfInfoConfig.serviceName	Perf-Info service name	NA	NA	NA
global.perfInfoConfig.host	Perf-Info service Host IP	NA	NA	NA
global.perfInfoConfig.port	Perf-Info service port	NA	NA	NA
global.perfInfoConfig.perfInfoRequestMap	Perf-Info service request endpoint	`/load`	NA	NA
global.http1PortName	The port name for http1 port exposed on ingressgateway	http	NA	NA
global.nfType	Configures the NF type using Helm charts for error detail content.	UDR	NA	NA
global.nfFqdn	Configures the NF FQDN using Helm charts for error detail content.	udr001.oracle.com	Valid fqdn	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	60s Unit: Seconds	NA	NA

3.1.13 ocudr-ingressgateway-prov Microservice Parameters

The following table provides parameters for ocudr-ingressgateway-prov microservice (API Gateway)

Table 3-14 ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
image.tag	Tag or version of Image	24.3.3	Not Applicable	NA
commonCfgClient.enabled	Flag to enable common configuration client	true	true/false	NA
commonCfgServer.configServerSvcName	Service name of common configuration service to which the client tries to poll for configuration updates	nudr-config	Valid server service name	NA
commonCfgServer.host	Host name of Common configuration server to which client tries to poll for configuration updates. This value is picked up if commonCfgServer.configServerSvcName is not available	10.75.224.123	Valid host details	NA
commonCfgServer.pollingInterval	Time interval between two subsequent polling requests from configuration client to server	5000	Valid period	NA
commonCfgServer.port	Port of Common Configuration server	5001	Valid Port	NA
commonServiceName	This is the common service name that is currently requesting for configuration updates from server	igw	Not Applicable	NA
connectionTimeout	Connection timeout limit configured on ingress gateway client towards the backend	10000	Unit: Milliseconds	NA
global.xfccHeaderValidation.validation.errorCauseOnValidationFailure	Configurable error cause to be returned when XFCC header validation fails at Ingress Gateway. This error is populated in ProblemDetails response in **ProblemDetails.cause** section	XFCC Validation Failed	NA	NA
global.xfccHeaderValidation.validation.errorCodeOnValidationFailure	Field determines the errorCode to be sent in response when XFCC header validation fails at Ingress Gateway	500	Valid http error code	NA

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.xfccHeaderValidation.validation.errorDescriptionOnValidationFailure	Configurable error description to be returned when XFCC header validation fails at Ingress Gateway. This error is populated in ProblemDetails response in **ProblemDetails.detail** section	Validation of SCP failed	NA	NA
global.xfccHeaderValidation.validation.errorTitleOnValidationFailure	Configurable error title to be returned when XFCC header validation fails at Ingress Gateway. This error is populated in ProblemDetails response in **ProblemDetails.title** section.	Internal Server Error	NA	NA
global.xfccHeaderValidation.validation.matchCerts	The number of certificates that need to be validated starting from the right most entry in the XFCC header	-1	(Value = -1 validation to be performed against all entries, value = +ve number, then validate from starting from the right most entry backwards)	NA
global.xfccHeaderValidation.validation.matchField	Field in XFCC header against which the configured nfList FQDN validation needs to be performed	Subject.CN	Valid XFCC header field	NA
global.xfccHeaderValidation.validation.nfList	List of configured NF FQDN's against which the matchField parameter present in the XFCC header of incoming request is validated	nfList • scp1.com • scp2.com	Valid url list received in Subject.CN if the matchField is Subject.CN	NA
global.xfccHeaderValidation.validation.redirectUrl	Redirect-Url string populated in LOCATION header and sent in response when XFCC header validation fails at Ingress Gateway and the value configured for errorCodeOnValidationFaulire lies in 3xx series	""	Valid redirect url	NA

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.xfccHeaderValidation.validation.retryAfter	Retry-after value in seconds populated in Retry-After header and sent in response, when XFCC header validation fails at Ingress Gateway and the value configured for errorCodeOnValidationFaulire lies in 3xx series	""	Retry-After value in seconds	NA
istioSidecarQuitUrl	Quit URL configured for side car	*istioQuitUrl	Not Applicable	This value is not to be changed unless it is different from the value configured in the global section reference
istioSidecarReadyUrl	Readiness URL configured for side car	*istioReadyUrl	Not Applicable	This value is not to be changed unless it is different from the value configured in the global section reference
maxConcurrentPushedStreams	Maximum number of concurrent pushed streams for Jetty client configuration	6000	Valid Number	NA
maxConnectionsPerDestination	Max connections allowed per destination from Ingress Gateway	10	Valid Number	NA
maxConnectionsPerIp	Maximum number of connections allowed from endpoint to other microservices	10	Valid Number	NA
maxRequestsQueuedPerDestination	Queue Size at the ocudr-ingressgateway pod	5000	Valid Number	NA
nfType	The NFType of the service producer Mandatory when oauthValidatorEnabled is true	UDR	Not Applicable	v

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nrfPublicKeyKubeNamespace	Namespace of the NRF publicKey secret Mandatory when oauthValidatorEnabled is true	ocudr	Not Applicable	NA
producerScope	Comma separated list of services hosted by service producer Mandatory when oauthValidatorEnabled is true	nudr-dr-prov,nudr-group-id-map-prov,slf-group-prov	Valid service list	NA
requestTimeout	Request timeout limit configured on ingress gateway client towards the backend	1000	Unit: Milliseconds	NA

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
routesConfig	Routes configured to connect to different microservices of UDR	routesConfig # configuring routes routesConfig: - id: traffic_mappi ng_http_prov uri: http://{{ .Re lease.Name }} -nudr-dr- provservice:5 001 path: / nudr-dr- prov/** order: 1 metadata: # oauthValidati on route level oauthValidato r: enabled: *oauthEnabled ForProv # xfccHeaderVal idation route level xfccHeaderVal idation: validationEna bled: *xfccValidati onEnabledProv - id: traffic_mappi ng_http_mgmt uri: http://{{ .Re	Not Applicable	NA

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		lease.Name }} -nudr-dr- provservice:5 001 path: / nudr-dr- mgm/** order: 2 metadata: # oauthValidati on route level oauthValidato r: enabled: *oauthEnabled ForProv # xfccHeaderVal idation route level xfccHeaderVal idation: validationEna bled: *xfccValidati onEnabledProv - id: traffic_mappi ng_http_group _prov uri: http://{{ .Re lease.Name }} -nudr-dr- provservice:5 001 path: / nudr-group- id-map- prov/** order: 3 metadata:		

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		# oauthValidation route level oauthValidator: enabled: *oauthEnabledForProv # xfccHeaderValidation route level xfccHeaderValidation: validationEnabled: *xfccValidationEnabledProv - id: traffic_mapping_http_slf_group_prov uri: http://{{ .Release.Name }}-nudr-dr-provservice:5001 path: /slf-group-prov/** order: 4 metadata: # oauthValidation route level oauthValidator: enabled: *oauthEnabledForProv		

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		# xfccHeaderValidation route level xfccHeaderValidation: validationEnabled: *xfccValidationEnabledProv - id: traffic_mapping_http_eir_prov uri: http://{{ .Release.Name }}-nudr-dr-provservice:5001 path: /n5g-eir-prov/** order: 5 metadata: # oauthValidation route level oauthValidator: enabled: *oauthEnabledForProv # xfccHeaderValidation route level xfccHeaderValidation: validationEnabled: *xfccValidati		

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

		onEnabledProv
		- id:
		traffic_mappi
		ng_http_dbcr_
		prov
		  uri:
		http://{{ .Re
		lease.Name }}
		-nudr-dbcr-
		auditor-
		service:5001
		  path: /
		nudr-dbcr/**
		  order: 6
		  metadata:
		  #
		oauthValidati
		on route
		level
		
		oauthValidato
		r:
		
		enabled:
		false
		  #
		xfccHeaderVal
		idation
		route level
		
		xfccHeaderVal
		idation:
		
		validationEna
		bled: false		

serviceMeshCheck	Flag to enable serviceMesh	*serviceMeshFlag	Not Applicable	This value is not to be changed unless it is different from the value configured in the global section reference

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.lciHeaderConfig.loadThreshold	If current load level is beyond the previously computed load level + loadThreshold, LCI headers are reported again.	40	NA	NA
global.lciHeaderConfig.localLciHeaderValidity	Validity period of the LCI headers reported to the consumer NF. LCI headers are reported again if the headers are reported as previously expired.	1000 (value in milliseconds)	NA	NA
global.lciHeaderConfig.producerSvcIdHeader	Header name, which holds the producer svc identity.	`custom_producer_svc_header`	NA	NA
global.ociHeaderConfig.enabled	Flag to enable or disable the OCI headers reporting	false	true or false	NA
global.ociHeaderConfig.producerSvcIdHeader	Header name, which holds the producer svc identity.	custom_producer_svc_header	NA	NA
global.ociHeaderConfig.validityPeriod	Validity period of the OCI headers reported to the consumer NF. OCI headers are reported again if the headers are reported as previously expired.	5000 (value in milliseconds)	NA	NA
global.ociHeaderConfig.overloadConfigRange.minor	Range to identify the minor overload condition	"[0-10]"	NA	NA
global.ociHeaderConfig.overloadConfigRange.major	Range to identify the major overload condition	"[10-70]"	NA	NA
global.ociHeaderConfig.overloadConfigRange.critical	Range to identify the critical overload condition	"[70-100]"	NA	NA
global.ociHeaderConfig.reductionMetrics.minor	Reduction metric to be reported for the minor overload condition	5 (Possible values 1 to 9 both inclusive)	NA	NA
global.ociHeaderConfig.reductionMetrics.major	Reduction metric to be reported for the major overload condition	10 (Possible values 5 to 15 both inclusive)	NA	NA
global.ociHeaderConfig.reductionMetrics.critical	Reduction metric to be reported for the critical overload condition	30 (Possible values 10 to 50 both inclusive)	NA	NA

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.nfInstanceId	NF instance Id of the producer NF	6faf1bbc-6e4a-4454-a507-a14ef8e1bc11	NA	NA
global.svcToSvcInstanceIdMapping.svcName	Back-end service name, which should match the producerSvcIdHeader value and perf info reported service name for the LCI or OCI headers reporting.	nf-registration	NA	NA
global.svcToSvcInstanceIdMapping.serviceInstanceId	Back-end service instance id to be included in the LCI or OCI headers	fe7d992b-0541-4c7d-ab84-c6d70b1b01b1	NA	NA
global.perfInfoConfig.pollingInterval	Configurable interval at which the load infromation is polled from the perf-info service at GW	5000	NA	NA
global.perfInfoConfig.serviceName	Perf-Info service name	NA	NA	NA
global.perfInfoConfig.host	Perf-Info service Host IP	NA	NA	NA
global.perfInfoConfig.port	Perf-Info service port	NA	NA	NA
global.perfInfoConfig.perfInfoRequestMap	Perf-Info service request endpoint	`/load`	NA	NA
global.lciHeaderConfig.enabled	Flag to enable or disable the LCI headers reporting	false	true or false	NA
global.http1PortName	The port name for http1 port exposed on ingressgateway	http	NA	NA
global.nfType	Configures the NF type using Helm charts for error detail content.	UDR	NA	NA
global.nfFqdn	Configures the NF FQDN using Helm charts for error detail content.	udr001.oracle.com	Valid fqdn	NA

Table 3-14 (Cont.) ocudr-ingressgateway-prov microservice (API Gateway) parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	This can be used to add custom annotation(s) to ingressgateway-prov specific deployment	`annotations:`
		
		`sidecar.istio.io/proxyCPU: "3000m"`
		
		`sidecar.istio.io/proxyCPULimit: "3000m"`
		
		`sidecar.istio.io/proxyMemory: "1Gi"`
		
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		
		`proxy.istio.io/config: |`
		
		`terminationDrainDuration: 60s`
			NA	NA

gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	60s	Unit: Seconds	NA

3.1.14 ocudr-egressgateway Microservice Parameters

Following table list parameters for ocudr-egressgateway microservice (API Gateway).

Table 3-15 ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
enabled	Flag to enable egress gateway	true	true/false	NA
image.name	Docker image name	ocegress_gateway	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
image.tag	Tag or version of image	24.3.3	NA	NA
image.pullPolicy	This setting indicates if image needs to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
initContainersImage.name	Docker image name	configurationinit	NA	NA
initContainersImage.tag	Tag or version of Image	24.3.3	NA	NA
initContainersImage.pullPolicy	This setting indicates if image needs to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
updateContainersImage.name	Docker Image name	configurationupdate	NA	NA
updateContainersImage.tag	Tag or version of Image	24.3.3	NA	NA
updateContainersImage.pullPolicy	This setting indicates if image needs to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
dbHookImage.name	Image name of dbHook	common_config_hook	NA	NA
dbHookImage.tag	Tag or version of dbHook Image	24.3.3	NA	NA
dbHookImage.pullPolicy	Pull policy of image	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
resources.limits.cpu	CPU allotment limitation	2	NA	NA
resources.limits.memory	Memory allotment limitation	2Gi	NA	NA
resources.limits.initServiceCpu	Maximum number of CPUs that Kubernetes allow the egress-gateway init container to use	1	NA	NA
resources.limits.initServiceMemory	Memory limit for egress-gateway init container	1Gi	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.limits.updateServiceCpu	Maximum number of CPUs that Kubernetes allow the egress-gateway update container to use	1	NA	NA
resources.limits.updateServiceMemory	Memory limit for egress-gateway update container	1Gi	NA	NA
resources.limits.commonHooksCpu	CPU limit for database Hook container	1	NA	NA
resources.limits.commonHooksMemory	Memory limit for database Hook container	1Gi	NA	NA
resources.requests.cpu	CPU allotment for ocudr-egressgateway pod	2	NA	NA
resources.requests.memory	Memory allotment for ocudr-egressgateway pod	2Gi	NA	NA
resources.requests.initServiceCpu	Number of CPUs that system guarantees for egress-gateway init container, and Kubernetes uses this value to decide the node to place the pod	1	NA	NA
resources.requests.initServiceMemory	Memory limit that system guarantees for egress-gateway init container, and Kubernetes uses this value to decide the node to place the pod	1Gi	NA	NA
resources.requests.updateServiceCpu	Number of CPUs that system guarantees for egress-gateway update container, and Kubernetes uses this value to decide the node to place the pod	1	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
resources.requests.updateServiceMemory	Memory limit that system guarantees for egress-gateway update container, and Kubernetes uses this value to decide the node to place the pod	1Gi	NA	NA
resources.requests.commonHooksCpu	Number of CPUs request for database Hook container, and Kubernetes uses this value to decide the node to place the pod	1	NA	NA
resources.requests.commonHooksMemory	Memory request for database Hook container and Kubernetes uses this value to decide on which node to place the pod.	1Gi	NA	NA
resources.target.averageCpuUtil	CPU utilization limit for auto scaling	80	NA	NA
service.ssl.tlsVersion	TLS version to be used	TLSv1.2	Valid TLS version	These are service fixed parameters
service.initialAlgorithm	Algorithm to be used. You can use ES256, but use its corresponding certificates	RS256	RS256/ES256	NA
service.ssl.privateKey.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.privateKey.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.privateKey.rsa.fileName	RSA private key stored in the secret	rsa_private_key_pkcs1.pem	NA	NA
service.ssl.privateKey.ecdsa.fileName	ECDSA private key stored in the secret	ecdsa_private_key_pkcs8.pem	NA	NA
service.ssl.certificate.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.certificate.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.certificate.rsa.fileName	RSA certificate stored in the secret	apigatewayrsa.cer	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.ssl.certificate.ecdsa.fileName	ECDSA certificate stored in the secret	apigatewayecdsa.cer	NA	NA
service.ssl.caBundle.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.caBundle.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.caBundle.fileName	caBundle file name stored in the secret	caroot.cer	NA	NA
service.ssl.keyStorePassword.k8SecretName	Name of the secret that store keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.keyStorePassword.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.keyStorePassword.fileName	keyStore password file name stored in the secret	key.txt	NA	NA
service.ssl.trustStorePassword.k8SecretName	Name of the secret which stores keys and certificates	ocudr-gateway-secret	NA	NA
service.ssl.trustStorePassword.k8NameSpace	Namespace in which secret is created	ocudr	NA	NA
service.ssl.trustStorePassword.fileName	trustStore password stored in the secret	trust.txt	NA	NA
maxUnavailable	Number of pods always running	24.3.3	NA	NA
minReplicas	Minimum number of replicas to scale to maintain an average CPU utilization	1	NA	NA
maxReplicas	Maximum number of replicas to scale to maintain an average CPU utilization	4	NA	NA
log.level.root	Log level to be shown on ocudr-egressgateway pod	WARN	valid level	NA
log.level.egress	Log level to be shown on ocudr-egressgateway pod for egress related flows	INFO	valid level	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
log.level.oauth	Logs level to be shown on ocudr-egressgateway pod for oauth related flows	INFO	valid level	NA
log.level.updateContainer	Log level for update container	INFO	valid level	NA
log.level.configclient	Log level for configuration client	INFO	valid level	NA
log.level.hook	Log level for hook	INFO	valid level	NA
fullnameOverride	Name to be used for deployment	ocudr-egressgateway	NA	This configuration is commented by default.
initssl	Flag to initialize SSL related infrastructure in init or update container	false	NA	NA
jaegerTracingEnabled	Flag to enable Jaeger Tracing	false	true/false	NA
openTracing.jaeger.udpSender.host	FQDN of Jaeger agent service	occne-tracer-jaeger-agent.occne-infra	Valid FQDN	NA
openTracing.jaeger.udpSender.port	UDP port of Jaeger agent service	6831	Valid Port	NA
openTracing.jaeger.probabilisticSampler	probabilisticSampler on Jaeger	0.5	Range: 0.0 - 1.0	Sampler makes a random sampling decision with the probability of sampling. For example if the value set is 0.1, approximately 1 in 10 traces are sampled.
enableOutgoingHttps	Flag to enable outgoing HTTPS requests	false	true or false	NA
istioSidecarQuitUrl	QuitUrl configurable for side car	*istioQuitUrl	Valid url	NA
istioSidecarReadyUrl	Readiness url configurable for side car	*istioReadyUrl	Valid url **Note:** The port used should be the admin configured for istio proxy container.	NA
serviceMeshCheck	Flag to enable serviceMesh	*serviceMeshFlag	true/false	

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
httpsTargetOnly	This is global parameter which will be taken into consideration if route (under routeConfig section) based httpsTargetOnly parameter is not available. #true: Select SbiRouting instances for https list only #false: Run existing logic as per provided scheme.	false	true/false	Double quotes to be enclosed for values of httpsTargetOnly
httpRuriOnly	This is global parameter which will be taken into consideration if route (under routeConfig section) based httpRuriOnly parameter is not available. #true: Means change Scheme of RURI to http #false: Keep scheme as is.	false	true/false	Double quotes to be enclosed for values of httpsRuriOnly
sbiRouting.sbiRoutingDefaultScheme	Default scheme applicable when 3gpp-sbi-target-apiroot header is missing	HTTPS	http/https	NA
sbiRouting.sbiRerouteEnabled	Set this flag to true if re-routing to multiple SCP instances is to be enabled.	false	true/false	NA
sbiRouting.peerConfiguration[0].id	Peer identifier for the peer	Value to be configured based on SCP details	NA	NA
sbiRouting.peerConfiguration[0].host	First SbiRouting instance HTTP IP/ FQDN	Value to be configured based on SCP details	NA	NA
sbiRouting.peerConfiguration[0].port	First SbiRouting instance Port	Value to be configured based on SCP details	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
sbiRouting.peerConfiguration[0].apiPrefix	First SbiRouting instance apiPrefix. Change this value to corresponding prefix if "/" is not expected to be provided along. Applicable only for SCP with TLS enabled.	Value to be configured based on SCP details	NA	NA
sbiRouting.peerConfiguration[0].virtualHost	This will have Http VirtualFQDN.	Value to be configured based on SCP details	NA	NA
sbiRouting.peerSetConfiguration[0].id	This is the peer set id that contains list of http and http instances	Value to be configured based on SCP details	NA	NA
sbiRouting.peerSetConfiguration[0].httpConfiguration[0].priority	This denotes the priority of the http instance that request needs to be forwarded. Lower the priority, higher the preference	Value to be configured based on SCP details	NA	NA
sbiRouting.peerSetConfiguration[0].httpConfiguration[0].peerIdentifier	This denotes the peer id that is present in the list of peers configured with unique ids	Value to be configured based on SCP details	NA	NA
sbiRouting.peerSetConfiguration[0].httpsConfiguration[0].priority	This denotes the priority of the https instance that request needs to be forwarded. Lower the priority, higher the preference	Value to be configured based on SCP details	NA	NA
sbiRouting.peerSetConfiguration[0].httpsConfiguration[0].peerIdentifier	This denotes the peer id that is present in the list of peers configured with unique ids	Value to be configured based on SCP details	NA	NA
headlessServiceEnabled	Enabling this will make the service type default to ClusterIP	false	true/false	NA
oauthClient.enabled	Flag to enable oauth client	false	true or false	Enable based on Oauth configuration

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

routesConfig	Valid routes to be configured for UDR egress traffic	routesConfig:
		# Route for NRF traffic
		- id: nrf_nfm_direct
		  uri: egress://dummy.dontchange
		  path: /nnrf-nfm/**
		  order: 1
		# Route for Notification traffic
		- id: notifications_direct
		  uri: http://dummy.dontchange
		  path: /**
		  order: 2
		  metadata:
		
		httpsTargetOnly: false
		
		httpRuriOnly: false
		
		sbiRoutingEnabled: false
		
		filterName1:
		  name: SbiRouting
		  args:
		
		peerSetIdentifier: set0
		
		customPeerSelectorEnabled: false
			Valid routes	The scpSetId to be configured based on the setId used for SCP route. These sets are configured under SCP section.

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

		errorHandling:
		  -
		errorCriteriaSet: notifications_direct_criteria_0
		
		actionSet: notifications_direct_action_0
		
		priority: 1
		
		filterNameControlShutdown:
		  name: ControlledShutdownFilter
		  args:
		
		applicableShutdownStates:
		  -
		"COMPLETE_SHUTDOWN"		

dnsSrv.host	Host of DNS Alternate Route Service	127.0.0.1	Valid host IP	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
dnsSrv.alternateRouteSvcName	Service name of Alternate Route Service. If Service name is provided, then this parameter is picked for integrating Egress Gateway with alternate route service. If IP or FQDN is expected to be provided then update this parameter as blank and update above parameter accordingly. If this parameter is populated with data then above parameter can be ignored	alternate-route	Valid service name of alternate route service	NA
dnsSrv.port	Port of DNS Alternate Route Service	80	Valid port	NA
dnsSrv.scheme	Scheme of request to be sent to alternate route service. By default it is HTTP	HTTP	HTTP/HTTPS	NA
dnsSrv.errorCodeOnDNSResolutionFailure	Configurable error code used in case of DNS resolution failure	425	Valid HTTP error code	NA
dnsSrv.errorDescriptionOnDNSResolutionFailure	Error description for DNS resolution failure. Populated in ProblemDetails response in ProblemDetails.detail section.	""	NA	NA
dnsSrv.errorCauseOnDNSResolutionFailure	Error cause for DNS resolution failure. Populated in ProblemDetails response in ProblemDetails.cause section.	""	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
dnsSrv.errorTitleOnDNSResolutionFailure	Error title for DNS resolution failure. Populated in ProblemDetails response in ProblemDetails.title section.	""	NA	NA
oauthClient.enabled	Flag to enable oauth client	false	true or false	Enable based on Oauth configuration
oauthClient.dnsSrvEnabled	Flag to enable DNS SRV for oAuth	false	true/false	NA
oauthClient.httpsEnabled	Flag to enable HTTPS support, which is a deciding factor for oauth request scheme and search query parameter in dns-srv request	false	true/false	NA
oauthClient.virtualFqdn	virtualFqdn value that needs to be populated and sent in dns-srv query	localhost:port	NA	Mandatory if oauthClient.dnsSrvEnabled is true
oauthClient.staticNrfList	List of Static NRF's	- localhost:port	NA	Mandatory if oauthClient.enabled is true
oauthClient.nfType	NF type of service consumer	UDR	NA	Mandatory if oauthClient.enabled is true
oauthClient.consumerPlmnMNC	MNC of service consumer	14	Valid MNC	NA
oauthClient.consumerPlmnMCC	MCC of service consumer	310	Valid MCC	NA
oauthClient.maxNonPrimaryNrfs	Maximum number of non-primary NRF instances to query based on retryErrorCodeSeries configured (from the list of non-primary NRF instances available) if a failure response is received from primary NRF	2	NA	Mandatory if oauthClient.enabled is true
oauthClient.apiPrefix	apiPrefix that needs to be appended in the Oauth request flow	""	Valid String	Mandatory if oauthClient.enabled is true

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
oauthClient.retryErrorCodeSeriesForSameNrf	Determines fallback condition to other non primary NRF instances if attempts for current NRF are exhausted and last received response from NRF matches retryErrorCodeSeries from errorSetId **(4XX, 5XX)** and errorCodes. If the configured attempts for current NRF are not exhausted and the previously received error matches retryErrorCodeSeries from errorSetId **(4XX, 5XX)** and errorCodes then retry same NRF again. errorCodes can be configured with specific set of errorCodes for a particular errorSetId, or can be configured as "-1", if configured "-1" then the entire range of errorCodes is considered for that particular errorSetId	`- errorSetId: 4XX errorCodes: 401, 402, 403, 404 - errorSetId: 5XX errorCodes: -1`	Valid http errors	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
oauthClient.retryErrorCodeSeriesForNextNrf		- errorSetId: 4XX errorCodes: 400, 401, 402, 403, 404 - errorSetId: 5XX errorCodes: -1	Valid HTTP errors	NA
oauthClient.retryExceptionListForSameNrf	List of available exceptions for oauth client while sending AccessToken requests to NRF. When an exception matches one of the exceptions in retryExceptionList, it tries next available NRF instance based on priority instead of retrying same NRF.	- TimeoutException - InterruptedException - ExecutionException - JsonProcessingException	Valid exception	NA
oauthClient.retryExceptionListForNextNrf		- TimeoutException - InterruptedException - ExecutionException - JsonProcessingException	Valid Exception	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
oauthClient.connectionTimeout	Determines the connection timeout in milliseconds for jetty client in case of subscription requests to NRF-Client Management Service and Access Token requests to NRF	1000 Unit: milliseconds	NA	Mandatory if oauthClient.enabled is true
oauthClient.requestTimeout	Determines the request timeout in milliseconds for jetty client for access token requests to NRF	1000 Unit: milliseconds	NA	Mandatory if oauthClient.enabled is true
oauthClient.attemptsForPrimaryNRFInstance	Determines the number of attempts available to query primary NRF	1	NA	Mandatory if oauthClient.enabled is true
oauthClient.attemptsForNonPrimaryNRFInstance	Determines the number of attempts available to query non-primary NRF	1	NA	Mandatory if oauthClient.enabled is true
oauthClient.defaultNRFInstance	Default NRF instance configuration. It is used to query access token from default NRF if access token is not successfully fetched from any of the available NRF instances	localhost:port	NA	NA
oauthClient.oauthClient.defaultErrorCode	Default error code to be sent in response to requesterNf when an error response from NRF is received for an AccessToken request.	503	Valid Error Code	Mandatory if oauthClient.enabled is true

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
oauthClient.nrfClientConfig.serviceName	The service name of NRF-Client Management Service used for constructing Subscription URL and sending out subscription request. When serviceName field is empty then nrfClientConfig.host is used in NRF-Client management service subscription request URL for host configuration	ocnf-client-nfmanagement	NA	NA
oauthClient.nrfClientConfig.host	The host configuration for NRF-Client management service for sending subscription requests when service name is not configured	10.75.224.123	Valid host ip/fqdn	NA
oauthClient.nrfClientConfig.port	The port configuration for NRF-Client management service for sending subscription requests	8080	Valid Port	NA
oauthClient.nrfClientConfig.nrfClientRequestMap	The request mapping URL for sending subscription requests from Egress-Gateway to NRF-Client management service	/v1/nrf-client/subscriptions/nrfRouteList	NA	NA
maxConcurrentPushedStreams	Maximum number of concurrent pushed streams for Jetty client configuration	1000	Valid Number	NA
maxConnectionsPerDestination	Max connections allowed per destination from Ingress Gateway	4	Valid Number	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
maxRequestsQueuedPerDestination	Queue Size at the ocudr-ingressgateway pod	5000	Valid Number	NA
maxConnectionsPerIp	Maximum number of connections allowed from endpoint to other microservices	10	Valid Number	NA
connectionTimeout	Connection timeout configured on ingress gateway client towards the backend	10000 Unit: Milliseconds	NA	NA
requestTimeout	Request timeout configured on ingress gateway client towards the backend	1000 Unit: Milliseconds	NA	NA
jettyIdleTimeout	Idle timeout allowed for Jetty in milliseconds	0	Unit: Milliseconds #(ms,<=0 -> to make timeout infinite)	NA
k8sServiceCheck	Flag to enable load balancing for egress instead of Kubernetes	false	true/false	NA
service.customExtension.labels	Custom labels that needs to be added to egress gateway specific service	null	NA	This can be used to add custom label(s) to egressgateway Service.
service.customExtension.annotations	Custom Annotations that needs to be added to egress gateway specific services	null	NA	This can be used to add custom annotation(s) to egressgateway Service.
deployment.customExtension.labels	Custom labels that needs to be added to egress gateway specific deployment	null	NA	This can be used to add custom label(s) to egressgateway Deployment.

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	This can be used to add custom annotation(s) to egressgateway deployment.	`annotations:`
		
		`sidecar.istio.io/proxyCPU: "1000m"`
		
		`sidecar.istio.io/proxyCPULimit: "1000m"`
		
		`sidecar.istio.io/proxyMemory: "1Gi"`
		
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		
		`proxy.istio.io/config: |`
		
		`terminationDrainDuration: 60s`
			NA	NA

readinessProbe.initialDelaySeconds	Configurable wait time before performing first readiness probe by the Kubelet	30 Unit: Seconds	NA	Do not change this value. If you see delays in pod to come up and probe is killing the pod then you should tune these parameters
readinessProbe.periodSeconds	Time interval for every readiness probe check	10 Unit: Seconds	NA	Do not change this value. If you see delays in pod to come up and probe is killing the pod then you should tune these parameters
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	NA	Do not change this default value

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after being failed	1	NA	Do not change this default value
readinessProbe.failureThreshold	When a pod starts and the probe fails, Kubernetes failureThreshold times before giving up	3	NA	NA
livenessProbe.initialDelaySeconds	Configurable wait time before performing the first liveness probe by Kubelet	30 Unit: Seconds	NA	Do not change this value. If you see delays in pod to come up and probe is killing the pod then you should tune these parameters
livenessProbe.periodSeconds	Time interval for every liveness probe check Note: Do not change this value. If you see delays in pod coming up and probe is killing the pod then you should consider tuning these parameters.	15 Unit: Seconds	NA	NA
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	NA	Do not change this default value
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after being failed	1	NA	Do not change this default value
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes tries failureThreshold times before giving up	3	NA	Do not change this default value

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainers	This configuration decides service level extraContainer support. The default value lets it to be dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA
commonCfgClient.enabled	Flag to enable common configuration client	true	true/false	NA
commonCfgServer.configServerSvcName	Service name of common configuration service to which the client tries to poll for configuration updates	nudr-config	Valid server service name	NA
commonCfgServer.host	Host name of Common configuration server to which client tries to poll for configuration updates. This value is picked up if commonCfgServer.configServerSvcName is not available	10.75.224.123	Valid host details	NA
commonCfgServer.port	Port number of Common Configuration server	5001	Valid port	NA
commonCfgServer.pollingInterval	Time interval between two subsequent polling requests from configuration client to server	5000	Valid period	NA
commonServiceName	The common service name that is currently requesting for configuration updates from server	egw	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
restoreBackupOnInstall	This flag when enabled picks up data from the backup table during installation of ingress gateway	false	true/false	NA
dropTablesOnUpgrade	This flag when enabled drops the common configuration tables on Helm upgrade if there is no data in the tables	false	true/false	NA
httpRuriOnly	This flag is enabled when the alternate route schema is set to HTTPS in nudr-nrf-client section	false	true/false	NA
autoRedirect	Flag to enable redirection in Egress Gateway. If enabled, Egress Gateway redirects to the URL present in location header when the response code is 301, 302, 303, 307, or 308.	false	true/false	NA
global.configurableErrorCodes.errorScenarios[i].exceptionType	Configurable exception or error type for an error scenario in Ingress Gateway. The only supported values are ConnectionTimeout, RequestTimeout, UnknownHostException, ConnectException, RejectedExecutionException, SSLHandshakeException, InternalError and NotFoundException,ClosedChannelException	Yes (If ConfigurableErrorCodes is enabled and as per need)	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
routeConfigMode	Mode of route configuration for sbiRouting. Possible values are HELM, REST	No	Helm	NA
configureDefaultRoute	Flag to configure default route. Configure this flag when routeConfigMode is configured as REST.	No	True	NA
dbConfig.backupDbName	configure when your backup table should have separate schema	No	NA	NA
componentValidator	custom validator configuration are needed for hooks to populate values in database	No	"com.oracle.common.egw.EgressCustomValidator"	NA
dependentValidators	It has comma separated values. for eg: "com.oracle.common.oauth.OauthCustomValidator, com.oracle.common.igw.IngressCustomValidator"	No	""	NA
isIpv6Enabled	Set this flag to true when deployed in Ipv6 enabled setup	No	false	NA
routesConfig[0].header	Provide the header name and regex (header predicate) in route to match header in the incoming request.	No	NA	NA
routesConfig[0].filterName2.exceptions	The type exceptions on which the re-route need to be attempted.	Yes (If sbiRerouteEnabled is true)	NA	NA
deploymentEgressGateway.imageTag	Image Tag name of egress gateway	No	1.14.0	NA
global.logStorage	Ephemeral storage configuration for log storage	No	0	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.crictlStorage	Ephemeral storage configuration for crictl storage	No	0	NA
nfSpecificConfig	This configuration is to enable or disable the NF specific default value configuration and a list of configuration files is defined in the default configuration list.	No	`nfSpecificConfig: enabled: false featureList: - logging - forwardheaderdetails - peerconfiguration - peersetconfiguration - routesconfiguration - messagelogging`	NA
honorDnsRecordttl	This configuration is to enable whether to honor the TTL values while resolving FQDN.	No	honorDnsRecordttl: false	NA
unusedDestinationCleanup	This configuration is used to enable scheduler which will remove unused FQDN after configured idle time.	No	unusedDestinationCleanup: false	NA
unusedDestinationCleanupAfter	This configuration is used to decide when to clean up unused destination.	No	unusedDestinationCleanupAfter: 20000 #(ms)	NA
unusedDestinationCleanupScheduler	This configuration is the time interval when the scheduler should run.	No	unusedDestinationCleanupScheduler: 600000 #(ms)	NA
ephemeralStorageLimit	Ephemeral storage limit	No	0	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
cncc.enablehttp1	This flag is used for enabling http1 traffic on Ingress Gateway. This need not be enabled unless you need to receive http1 traffic on Ingress Gateway	false	true/false	NA
userAgentHeaderValidationConfigMode	This flag is used to accept user-agent configurations from HELM or REST	HELM	HELM/REST	NA
userAgentHeaderValidation.enabled	This flag is used to enable the user-agent feature	false	true/false	NA
userAgentHeaderValidation.consumerNfTypes	Use this parameter to confgure who is consuming UDR from different NF Types, example like PCF,NRF,UDM	Empty	All NF Types that are defined in 3gpp specification. Example PCF,NRF,NEF,UDM	NA
serverHeader.autoBlacklistProxy.enabled	To enable blacklisting in server header feature	No	NA	default is false
serverHeader.autoBlacklistProxy.errorCodeList	List of error codes to check for blacklisting	No	NA	NA
serverHeader.autoBlacklistProxy.blacklistingPeriod	Time to blacklist a peer in milliseconds	No	NA	NA
userAgentHeaderConfigMode	parameter to configure mode of configuration	No	HELM	NA
userAgentHeader.enabled	flag to enable or disable the feature	No	False	NA
userAgentHeader.nfType	consumer nf type	No	empty string	NA
userAgentHeader.nfInstanceId	instance id of consumer nf type	No	empty string	NA
userAgentHeader.addfqdnToHeader	flag to indicate if user agent header should be generated with fqdn	No	false	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
userAgentHeader.overwriteHeader	flag to indicate if user agent header should be overwritten if already present in the request	No	false	NA
userAgentHeader.nfFqdn	fqdn of nf	No	empty string	NA
routesConfig[0].filterName2.withServerHeaderSupport.retries	Number of re-routes to be attempted to alternate SCP instances if request matches this route's path in server header scenario	No	NA	NA
routesConfig[0].filterName2.withServerHeaderSupport.methods	The type of methods for which the re-route need to be attempted in server header scenario	No	NA	NA
routesConfig[0].filterName2.withServerHeaderSupport.statuses	The type response error codes on which the re-route need to be attempted in server header scenario	No	NA	NA
routesConfig[0].filterName2.withServerHeaderSupport.exceptions	The type exceptions on which the re-route need to be attempted in server header scenario	No	NA	NA
routesConfig[0].metadata.serverHeaderEnabled	To enable or Disable server header feature	No	NA	default is false
routesConfig[0].metadata.serverHeaderNFtypes	List of NF types for server header	No	NA	NA
global.lciHeaderConfig.enabled	Flag to enable LCI Header	False	True or False	NA
global.lciHeaderConfig.loadThreshold	Egress Gateway decides whether or not to send the header in the next response using this threshold value	40	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.lciHeaderConfig.localLciHeaderValidity	Used to send the LCI Header response for each specific interval. If it is 60000 ms for every 60 sec, this response is sent	1000	NA	NA
global.lciHeaderConfig.consumerSvcIdHeader	Using this header, EGW identifies the destination address details and based on that, it decides whether or not to send the lci header	ConsumerHeader	NA	NA
global.lciHeaderConfig.producerSvcIdHeader	This header is configured in the same way in which the header from the response of back-end service (nudr-drservice) is received	UDRHeader	NA	NA
global.ociHeaderConfig.enabled	Flag to enable OCI Header	False	True or False	NA
global.ociHeaderConfig.consumerSvcIdHeader				
global.ociHeaderConfig.producerSvcIdHeader	This header is configured in the same way in which the header from the response of back-end service (nudr-drservice) is received.	UDRHeader	NA	NA
global.ociHeaderConfig.validityPeriod	Used to send OCI Header in the response for each specific interval. If it is 30000 ms for every 30 sec, this response is sent.	5000	NA	NA
global.ociHeaderConfig.overloadConfigRange	The back-end service load level is determined based on this configuration range	minor: "[0-80]" major: "[80-85]" critical: "[90-100]"	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.ociHeaderConfig.reductionMetrics	Indicates the percentage value in which we will give information to consumer that how much traffic will be dropped before sending.	minor: 5 #(Possible values 1 to 9 both inclusive) major: 15 #(Possible values 5 to 15 both inclusive) critical: 25 #(Possible values 10 to 50 both inclusive)	NA	NA
global.nfInstanceId	Indicates the global value being used at the NF level.	*nfInsId	NA	NA
global.svcToSvcInstanceIdMapping.svcName	Indicates the back-end service name, which is configured. This value should match with the service in which perf-info reports the load.	ocudr-nudr-drservice	NA	NA
global.svcToSvcInstanceIdMapping.serviceInstanceId	This value is unique for the service and it needs to be configured with the same value in which NFProfile (for nudr-drservice) for the svc level is sent.	ae870316-384d-458a-bd45-025c9e748976	NA	NA
global.perfInfoConfig.pollingInterval	Configurable interval at which load infromation is polled from the perf-info service at Gateway.			
global.perfInfoConfig.serviceName	Perf-Info service name	NA	NA	NA
global.perfInfoConfig.host	Perf-Info service Host IP	NA	NA	NA
global.perfInfoConfig.port	Perf-Info service port	NA	NA	NA
global.perfInfoConfig.perfInfoRequestMap	Perf-Info service request endpoint	`/load`	NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
deDupeResponseHeader	Used for handling the duplicate values in the response headers. Multiple values can be provided with a space. For example: content-type nettylatency requestmethod, RETAIN_LAST	False	True or False	NA
oauthEnabled	To enable or disable oauth at the route level	False	True or False	NA
http1.enableOutgoingHTTP1	Flag to enable or disable the outgoing http1.1 traffic for Egress Gateway	False	True or False	NA
sbiroutingerrorcriteriasets.id	Unique id for sbiRoutingErrorCriteriaSet		NA	NA
sbiroutingerrorcriteriasets.method	Methods for which a reroute or retry is triggered		NA	NA
sbiroutingerrorcriteriasets.response.statuses.statusSeries	Http Status series for which a reroute or retry is triggered when an error response from downsream is received		NA	NA
sbiroutingerrorcriteriasets.response.statuses.status	Specific HTTP status that belongs to statusSeries for which a reroute or retry has to be triggered. To enable a retry or reroute for all the HTTP status belonging to statusSeries, configure as **-1**		NA	NA
sbiroutingerrorcriteriasets.response.exceptions	Specific exceptions for which a reroute or retry is triggered		NA	NA

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
sbiroutingerrorcriteriasets.response.headersMatchingScript	A comma separated String values in the following format: 1st token: **headerCheck** (Hard coded values) 2nd to n-1 token: Header names that need to be validated nth token: regex expression for the header validation **Note:** The final result would be an aggregated **OR** of the individual header checks.		NA	NA
sbiroutingerroractionsets.id	Unique Id for sbiRoutingErrorActionSet		NA	NA
sbiroutingerroractionsets.action	Action that needs to be taken when a specific criteria set is matched. Currently, only 2 values are supported: **reroute/retry**		NA	NA
sbiroutingerroractionsets.attempts	Maximum number of retries to either same or different peer in case of error or failures from the back-end.		NA	NA
sbiroutingerroractionsets.blackList.enabled	Flag to enable the peer blacklist feature using the server headers received in the response.		NA	NA
sbiroutingerroractionsets.blackList.duration	The duration for which the peer is blacklisted and no traffic is routed to that peer for this period.		NA	

Table 3-15 (Cont.) ocudr-egressgateway microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
http1.enableOutgoingHTTP1	Flag to enable or disable outgoing http1.1 traffic for Egress Gateway		NA	NA
dnsRefreshDelay	Dns Refresh Delay in milli-seconds	10000 Unit: Milliseconds	NA	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	60s Unit: Seconds	NA	NA

3.1.15 alternate-route Microservice Parameters

Following table list parameters for alternate-route microservice.

Table 3-16 alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainers	This configuration decides service level extraContainer support. The default value lets it to be dependent on what is configured in the global level	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • DISABLED • USE_GLOBAL_VALUE	NA
replicaCount	Indicates the number of pods that need to be created as part of deployment	2	Any valid number from 1 to 4	NA
deploymentDnsSrv.image	Name of the docker image of Alternate Route application	alternate_route	NA	NA
deploymentDnsSrv.tag	Tag or version of the image of Alternate Route application	24.3.3	NA	NA
deploymentDnsSrv.pullPolicy	This setting indicates if image need to be pulled or not	IfNotPresent	Possible Values: Always IfNotPresent Never	NA
dbHookImage.name	Image name of dbHook	common_config_hook	Not Applicable	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
dbHookImage.tag	Tag or version of dbHook image	24.3.3	Not Applicable	NA
dbHookImage.pullPolicy	Pull policy of image	IfNotPresent	Possible Values: Always IfNotPresent Never	NA
service.type	Type of Kubernetes service	ClusterIP	Possible Values: ClusterIP NodePort LoadBalancer	NA
service.customExtension.labels	Custom labels that needs to be added to alternate-route Kubernetes service	null	NA	NA
service.customExtension.annotations	Custom Annotations that needs to be added to alternate-route Kubernetes service	null	NA	NA
deployment.customExtension.labels	Custom labels that needs to be added to alternate-route Kubernetes deployment	null	NA	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes

deployment.customExtension.annotations	his can be used to add custom annotation(s) to egressgateway deployment.	`annotations:`
		
		`sidecar.istio.io/proxyCPU: "1000m"`
		
		`sidecar.istio.io/proxyCPULimit: "1000m"`
		
		`sidecar.istio.io/proxyMemory: "1Gi"`
		
		`sidecar.istio.io/proxyMemoryLimit: "1Gi"`
		
		`proxy.istio.io/config: |`
		
		`terminationDrainDuration: 60s`
			NA	NA

minReplicas	Minimum number of replicas to scale to maintain an average CPU utilization	1	NA	NA
maxReplicas	Maximum number of replicas to scale to maintain an average CPU utilization	5	NA	NA
ports.servicePort	Port number of Kubernetes service	80	Valid Port	NA
ports.containerPort	Container port number that represents a network port in a single container	8004	Valid Port	NA
ports.actuatorPort	Actuator Port number	9090	Valid Port	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
log.level.root	Log level for root logs	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.altroute	Log level for alternate route logs	WARN	Possible Values: WARN INFO DEBUG	NA
log.level.configclient	Log level for configuration client logs	WARN	Possible Values: WARN INFO DEBUG	NA
coherence.port	Coherence port	No	8000	NA
coherence.messagingPort1	Port used by Coherence for communication	No	8095	NA
coherence.messagingPort2	Port used by Coherence for communication	No	8096	NA
log.level.hook	Log level for hook logs	warn	Possible Values: WARN INFO DEBUG	NA
staticVirtualFqdns[0].name	Name of the virtual FQDN	NA	Valid name: ex: http://abc.test.com	NA
staticVirtualFqdns[0].alternateFqdns[0].target	Name of the alternate FQDN mapped to above virtual FQDN	NA	target FQDN ex: abc.test.com	NA
staticVirtualFqdns[0].alternateFqdns[0].port	Port number of the alternate FQDN	NA	Valid Port	NA
staticVirtualFqdns[0].alternateFqdns[0].priority	Priority number of the alternate FQDN	NA	Priority Number	NA
dnsSrvEnabled	Flag to enable DNS-SRV query to coreDNS Server	true	true/false	NA
dnsSrvFqdnSetting.enabled	Flag to enable the usage of custom pattern for FQDN while triggering DNS-SRV query	true	true/false	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
dnsSrvFqdnSetting.pattern	Flag to enable the usage of custom pattern for FQDN while triggering DNS-SRV query	"_{scheme}._tcp.{fqdn}."	NA	NA
asynTaskExecutor.corePoolSize	This is the number of threads available in the "registration" thread pool which performs registration asynchronously	10	Valid number	NA
asynTaskExecutor.maxPoolSize	This is the limit for maximum number of threads in the "registration" thread pool	10	Valid number	NA
asynTaskExecutor.queueCapacity	This is the limit for number of requests that get queued for the availability of thread from "registration" thread pool	100	Valid number	NA
refreshTaskExecutor.corePoolSize	This is the number of threads available in the "auto-refresh" thread pool that performs "auto-refresh" scheduled task	10	Valid number	NA
refreshTaskExecutor.maxPoolSize	This is the limit for maximum number of threads in the "auto-refresh" thread pool	10	Valid number	NA
refreshTaskExecutor.queueCapacity	Limit for number of requests that get queued for the availability of thread from "auto-refresh" thread pool	20	Valid number	NA
cleanupTaskExecutor.corePoolSize	Number of threads available in the "auto-cleanup" thread pool which performs "auto-cleanup" scheduled task	3	Valid number	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
cleanupTaskExecutor.maxPoolSize	Limit for maximum number of threads in the "auto-cleanup" thread pool	3	Valid number	NA
cleanupTaskExecutor.queueCapacity	Limit for number of requests that get queued for the availability of thread from "auto-cleanup" thread pool	5	Valid number	NA
refreshScheduler.enabled	Flag to enable the "auto-refresh" scheduled task	true	true/false	NA
refreshScheduler.interval	Use this to trigger "refresh-scheduler: as per time interval configured in minutes	60	Unit: Minutes	NA
refreshScheduler.auditorShuffle	Flag to enable auditor (auto-refresh) functionality to randomly rotate among available non-leader pods	false	true/false	NA
refreshScheduler.throttling.burstLimit	Limit for number of DNS-SRV queries that can be sent in single loop while doing auto-refresh. It should always be greater than zero	50	Valid Number	NA
refreshScheduler.throttling.burstInterval	Time to wait (in seconds) before triggering the next burst of DNS-SRV queries while doing auto-refresh. This should always be greater than zero	5	Unit: Seconds	NA
refreshScheduler.throttling.threadPoolSize	Number of threads in ThreadPoolExecutors that schedules the auto refresh bursts as per burstInterval	5	Valid Number	NA
cleanupScheduler.enabled	Flag to enable the "auto-cleanup" scheduled task	true	true/false	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
cleanupScheduler.interval	Use to trigger "cleanup-scheduler" as per time interval configured in minutes	100	Unit: Minutes	NA
cleanupScheduler.lastUsedInterval	This time interval in minutes is used for selection of records for cleanup	200	Unit: Minutes	NA
readinessProbe.initialDelaySeconds	Configurable wait time before performing the first readiness probe by the Kubelet	30	Unit: Seconds	Do not change this value. If you see delays in pod to come up and probe is killing the pod then tune these parameters
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	Unit: Seconds	Do not change this default value.
readinessProbe.periodSeconds	Time interval for every readiness probe check	10	Unit: Seconds	Do not change this value. If you see delays in pod to come up and probe is killing the pod then tune these parameters
readinessProbe.successThreshold	Number of seconds after which the probe times out	1	NA	Do not change this default value.
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes tries failureThreshold times before giving up	3	NA	Do not change this default value
livenessProbe.initialDelaySeconds	Configurable wait time before performing the first liveness probe by the Kubelet	30	Unit: Seconds	Do not change this value. If you see delays in pod to come up and probe is killing the pod then tune these parameters
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	3	Unit: Seconds	Do not change this default value

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
livenessProbe.periodSeconds	Time interval for every liveness probe check	15	Unit: Seconds	Do not change this value. If you see delays in pod to come up and probe is killing the pod then tune these parameters
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	1	NA	Do not change this default value.
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes tries failureThreshold times before giving up	3	NA	Do not change this default value
resources.limits.cpu	CPU Limit	2	NA	NA
resources.limits.memory	Memory Limit	2Gi	NA	NA
resources.limits.commonHooksCpu	CPU limit of database Hook container	1	Not applicable	NA
resources.limits.commonHooksMemory	Memory limit of database Hook container	1Gi	Not applicable	NA
resources.requests.cpu	Number of CPUs requested	1	NA	NA
resources.requests.memory	Memory Requested	2Gi	NA	NA
resources.requests.commonHooksCpu	Number of CPU request for database Hook container. Kubernetes uses this value to decide the node to place the pod	1	Not Applicable	NA
resources.requests.commonHooksMemory	Memory request for database Hook container. Kubernetes uses this value to decide the node to place the pod	1Gi	Not Applicable	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
target.averageCpuUtil	Limit of CPU utilization (in percentage) of the Pod for auto scaler to pitch-in	80	Unit: Percent	NA
jaegerTracingEnabled	Flag to enable Jaeger tracing	false	true/false	NA
openTracing.jaeger.udpSender.host	Jaeger Host	occne-tracer-jaeger-agent.occne-infra	Valid host details	NA
openTracing.jaeger.udpSender.port	Jaeger Port number	6831	Valid Port	NA
openTracing.probablisticSampler	Probabilistic Sampler on Jaeger	0.5	Range: 0.0 - 1.0	Sampler makes a random sampling decision with the probability of sampling. For example: If the value set is 0.1, approximately 1 in 10 traces will be sampled.
istioSidecarQuitUrl	Quit URL configurable for side car. Change only if the URL is different for the side car container in this microservice	*istioQuitUrl	NA	Do not change this value unless it is different from the value configured in global section reference
istioSidecarReadyUrl	Readiness URL configurable for side car. Change only if the URL is different for the side car container in this microservice	*istioReadyUrl	Not Applicable	Do not change this value unless it is different from the value configured in global section reference
serviceMeshCheck	Flag to enable service mesh. Set this to false when side car is not included for this service	*serviceMeshFlag	Not Applicable	Do not change this value unless it is different from the value configured in global section reference
commonCfgClient.enabled	Flag to enable persistent configuration	true	true/false	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
commonCfgServer.configServerSvcName	Service name of common configuration service to which the client tries to poll for configuration updates	nudr-config	Valid server service name	NA
commonCfgServer.host	Host name of Common configuration server to which client tries to poll for configuration updates. This value is picked up if commonCfgServer.configServerSvcName is not available	10.75.224.123	Valid host details	NA
commonCfgServer.port	Port of Common Configuration server	5001	Valid Port	NA
commonCfgServer.pollingInterval	Interval between two subsequent polling requests from configuration client to server	5000	Valid period	NA
commonServiceName	This is the common service name that is currently requesting for configuration updates from server	alt-route	Not Applicable	NA
restoreBackupOnInstall	This flag when enabled picks up the data from the backup table during installation of ingress gateway	false	true/false	NA
dropTablesOnUpgrade	This flag when enabled drops the common configuration tables on Helm upgrade if there is no data in the tables	false	true/false	NA
global.metricPrefix	NF level metric prefix that is added to all the metrics	No	empty string	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.metricSuffix	NF level metric suffix that is added to all the metrics	No	empty string	NA
metricPrefix	Service level metric prefix that is added to all the metrics	No	empty string	NA
metricSuffix	Service level metric suffix that is added to all the metrics	No	empty string	NA
minAvailable	Number of Pods must always be available, even during a disruption.	Yes	0	This parameter is used for PodDisruptionBudget Kubernetes resource
dbConfig.backupDbName	Configure when your backup table should have separate schema	No	NA	NA
global.logStorage	Ephemeral storage configuration for log storage	No	0	NA
global.crictlStorage	Ephemeral storage configuration for crictl storage	No	0	NA
ephemeralStorageLimit	Ephemeral storage limit	No	0	NA
tolerationsSetting	Flag to enable the toleration setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value can be used. If set to ENABLED tolerations configuration below in the same section is used.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelection	Flag to enable the nodeSelection setting at the microservice level or global level. If set to USE_GLOBAL_VALUE global.tolerations configured value is used. If set to ENABLED tolerations configuration below in the same section is used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: • ENABLED • USE_GLOBAL_VALUE • DISABLED	NA

tolerations	When tolerationsSetting is ENABLED, configure tolerations here.	[]	Example:
			`tolerations:`
			`- key: "key1"`
			`operator: "Equal"`
			`value: "value1"`
			`effect: "NoSchedule"`
				NA

helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: • v1 • v2	NA
nodeSelector.nodeKey	NodeSelector key configuration at the microservice level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	''	Valid key for a label for a particular node	NA

Table 3-16 (Cont.) alternate-route microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelector.nodeValue	NodeSelector Value configuration at the global level when helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. And this configuration does not depend on nodeSelection flag, once configured this is used for all microservices.	' '	Valid value pair for the above key for a label for a particular node.	NA
nodeSelector	NodeSelector configuration when helmBasedConfigurationNodeSelectorApiVersion is set to v2. Uncomment and use this configuration when required. Else keep this commented.	{}	Valid key value pair matching a node for nodeSelection by a pod.	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	60s	Unit: Seconds	NA

3.1.16 Database Resolution Auditor Service

The following table list the user parameters for database resolution auditor service.

Table 3-17 Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
enabled	Flag for enable or disable nudr-notify-service.	false	true or false
image.name	Docker image name.	nudr_dbcr_auditor_service	NA
image.tag	Docker image tag.	24.3.0	NA
replicas	replicas of nudr_dbcr_auditor_service.	1	NA

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
service.type	nudr_dbcr_auditor_service service type.	ClusterIP	The Kubernetes service type for exposing UDR deployment: ClusterIP NodePort LoadBalancer
minReplicas	Minimum replicas Minimum number of pods.	1	NA
maxReplicas	Maximum replicas Maximum number of pods.	1	NA
maxUnavailable	Number of replicas that can go down during a disruption. This parameter is used for PodDisruptionBudget k8s resource.	1	NA
service.port.http	HTTP port The HTTP port to be used in nudr_dbcr_auditor_service.	5001	NA
service.port.https	HTTPS port The HTTP port to be used in nudr_dbcr_auditor_service.	5002	NA
service.port.management	Management port. The actuator management port to be used for nudr_dbcr_auditor_service.	9000	NA
deployment.replicaCount	Replicas of nudr_dbcr_auditor_service pod. Number of nudr_dbcr_auditor_service pods to be maintained by replica set created with the deployment.	1	NA
hikari.poolsize	Mysql Connection pool size The hikari pool connection size to be created at start up.	10	NA
hikari.connectionTimeout	MYSQL connection timeout	1000	Unit: Milliseconds

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
hikari.minimumIdleConnections	Minimum idle MYSQL connections maintained	5	Valid number is less than poolsize configured
hikari.idleTimeout	Idle MYSQL connections timeout	10000 Unit: Milliseconds	NA
hikari.queryTimeout	Query timeout for a single database query. If set to -1, there will be no timeout set for database queries.	-1 Unit: Seconds	NA
logging.level.root	Log Level Log level of the export tool pod.	WARN	Possible Values - WARN INFO DEBUG
resources.requests.cpu	CPU Allotment for nudr-notify-service pod. The CPU to be allocated for notify service pod during deployment.	2	NA
resources.requests.memory	Memory allotment for nudr-bulk-import pod. The memory to be allocated for nudr-bulk-import pod during deployment.	2Gi	NA
resources.requests.logStorage	Log storage for ephemeral storage allocation request.	50 Unit: MB	NA
resources.requests.crictlStorage	Crictl storage for ephemeral storage allocation request.	2 Unit: MB	NA
resources.limits.cpu	CPU allotment limitation.	2	NA
resources.limits.memory	Memory allotment limitation.	2Gi	NA
resources.limits.logStorage	Log storage for ephemeral storage allocation limit.	1000 Unit: MB	NA
resources.limits.crictlStorage	Crictl storage for ephemeral storage allocation limit.	2 Unit: MB	NA
resources.target.averageCpuUtil	CPU utilization limit for autoscaling. CPU utilization limit for creating HPA.	80	NA

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
startupProbe.failureThreshold	Configurable number of times startup probe will be retried for failures. **Note:** Do not change this value. If there is delay in pod coming up and probe is killing the pod, then you must adjust these parameters.	40 Unit: Seconds	NA
startupProbe.periodSeconds	Time interval for every startup probe check. **Note:** Do not change this value. If we see delays in pod coming up and probe is killing the pod then we should consider tuning these parameters.	5	Unit: Seconds
extraContainers	Configuration to enable debug tool container	DISABLED	Disabled or enabled
serviceMeshCheck	Enable when deployed in serviceMesh, referred to the serviceMeshCheck in global section. Set to false when side car is not included for this service.	*serviceMeshFlag	NA Note: Do not change this value unless it is different from the value configured in global section reference.
istioSidecarReadyUrl	Readiness URL configurable for sidecar, refer to the global section istioSidecarReadyUrl. Change only if the URL is different for the sidecar container in this microservice.	*istioReadyUrl	NA Note: Do not change this value unless it is different from the value configured in global section reference.
tolerationsSetting	Flag to enable micro service level or global level toleration setting. If set to USE_GLOBAL_VALUE global.tolerations then configured value will be used. If set to ENABLED, service specific tolerations setting will be used.	USE_GLOBAL_VALUE	Allowed Values: ENABLED D USE_GLOBAL_VALUE DISABLE

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
nodeSelection	Flag to enable micro service level or global level nodeSelection setting. If set to USE_GLOBAL_VALUE global.tolerations configured value will be used. If set to ENABLED, service specific tolerations setting will be used. This is applicable for v2 apiVersion nodeSelector configuration.	USE_GLOBAL_VALUE	Allowed Values: ENABLED D USE_GLOBAL_VALUE DISABLE
tolerations	Configure the tolerations when tolerationsSetting is ENABLED.	[]	NA
helmBasedConfiguration NodeSelectorApiVersion	Node Selector API version setting	v2	Allowed Values: • v1 • v2
nodeSelector	NodeSelector configuration when helmBasedConfiguration NodeSelectorApiVersion is set to v2. For v1 you can have key:value pair configured.	{}	Valid key value pair matching a node for nodeSelection by a pod. v1 Example: zone: Antartica
dbConflictResolution.auditException	The list of exception tables to be monitor.	The following configuration can be used as default value if deployed in EIR mode. ID1TODATA$EX,ID2TODATA$EX,ID3TODATA$EX,ID4TODATA$EX,ID5TODATA$EX,ID6TODATA$EX,ID7TODATA$EX,PROFILE_DATA$EX If installed in SLF, UDR, or All modes change the value as following: ID1TODATA$EX,ID2TODATA$EX,ID3TODATA$EX,ID4TODATA$EX,ID5TODATA$EX,PROFILE_DATA$EX	NA

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
dbConflictResolution.deleteException	The list of exception tables to be cleared without auditing.	All mode: GROUP_ID_MAPEX,SLF_GROUP_NAMEEX,PCF_DATA$EX,SUBSCRIPTION$EX,NEF_DATA$EX,EXPOSURE_DATA$EX,NEF_SUBSCRIPTION$EX,AUTHENTICATION_DATA$EX,UDM_AUTH_ENC$EX,CONTEXT_DATA_1$EX,CONTEXT_DATA_2EX,UDM_DATA1EX,UDMSUBEX,UDM_DATA2EX,UE_UPDATE_CONFIRMATION_DATA$EX,UE_SPECIFIC_DATA_1$EX,UE_SPECIFIC_DATA_2EX,UDM_SHARED_DATAEX,PROVISIONED_DATA_1$EX,UDM_SUBSCRIPTION$EX,PCF_AMDATA$EX,PCF_UEPOLICYSET$EX You can configure the following values based on deployment mode: • UDR Mode: PCF_DATA$EX,SUBSCRIPTION$EX,NEF_DATA$EX,EXPOSURE_DATA$EX,NEF_SUBSCRIPTION$EX,AUTHENTICATION_DATA$EX,UDM_AUTH_ENC$EX,CONTEXT_DATA_1$EX,CONTEXT_DATA_2EX,UDM_DATA1EX,UDMSUBEX,UDM_DATA2EX,UE_UPDATE_CONFIRMATION_DATA$EX,UE_SPECIFIC_DATA_1$EX,UE_SPECIFIC_DATA_2EX,UDM_SHARED_DATAEX,PROVISIONED_DATA_1$EX,UDM_SUBSCRIPTION$EX,PCF_AMDATA$EX,PCF_UEPOLICYSET$EX • SLF Mode: GROUP_ID_MAP$	NA

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
		EX,SLF_GROUP_NAME$EX • EIR Mode: EIR_DATA$EX	
dbConflictResolution.deleteConsumer	The list of consumer data tables to be audited.	All mode: PROFILE_DATA,GROUP_ID_MAP,PCF_DATA,NEF_DATA,EXPOSURE_DATA,AUTHENTICATION_DATA,UDM_AUTH_ENC,CONTEXT_DATA_1,CONTEXT_DATA_2,UDM_DATA1 UDMSUB,UDM_DATA2,UE_UPDATE_CONFIRMATION_DATA,UE_SPECIFIC_DATA_1,UE_SPECIFIC_DATA_2,PROVISIONED_DATA_1,EIR_DATA,PCF_AMDATA,PCF_UEPOLICYSET You can configure the following values based on deployment mode: • EIR mode: PROFILE_DATA,EIR_DATA • SLF mode: PROFILE_DATA,GROUP_ID_MAP • UDR mode: PROFILE_DATA,PCF_DATA,NEF_DATA,EXPOSURE_DATA,AUTHENTICATION_DATA,UDM_AUTH_ENC,CONTEXT_DATA_1,CONTEXT_DATA_2,UDM_DATA1 UDMSUB,UDM_DATA2,UE_UPDATE_CONFIRMATION_DATA,UE_SPECIFIC_DATA_1,UE_SPECIFIC_DATA_2,PROVISIONED_DATA_1,PCF_AMDATA,PCF_UEPOLICYSET	NA

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
dbConflictResolution.auditIdxtodata	The list of data id tables to be audited.	EIR mode: ID1TODATA,ID2TODATA,ID3TODATA,ID4TODATA,ID5TODATA,ID6TODATA,ID7TODATA You can configure the following values based on deployment mode: All/SLF/UDR modes: ID1TODATA,ID2TODATA,ID3TODATA,ID4TODATA,ID5TODATA	NA
mateSitesIgwIPList	It is a comma separated list of mate site Ingress gateway provisioning IPs or FQDN with port. Example: In case of three-site setup installed on myudr1, myudr2, and myudr3 namespaces and the release name used is ocudr. The configuration must be same as the default value for site1 installed on myudr1 namespace. For site2 installed on myudr2 namespace, the value must be `http://ocudr-ingressgateway-prov.myudr:80,http://ocudr-ingressgateway-prov.myudr3:80` so that site2 has the details of its mate sites. The site3 installed on myudr3 namespace, the value should be `http://ocudr-ingressgateway-prov.myudr:80,http://ocudr-ingressgateway-prov.myudr2:80`, so that site3 has the details of its mate sites.	`http://ocudr-ingressgateway-prov.myudr2:80,http://ocudr-ingressgateway-prov.myudr3:80`	Valid IP with port or FQDN with port. This must be comma separated.
gracefulShutdown.gracePeriod	The graceful shutdown period used for Diameter Gateway service termination.	30s Unit Seconds	NA

Table 3-17 (Cont.) Database Resolution Auditor Service

Parameter	Description	Default value	Range or Possible Values (If applicable)
dbcrAuditInterval	Specifies the database audit interval. This is an engineering configuration, it is recommended to not modify this value.	900000 Unit: Milliseconds	NA

4
Upgrading UDR

This chapter provides information about upgrading Oracle Communications Cloud Native Core, Unified Data Repository (UDR) deployment to the latest release.

4.1 Supported Upgrade Paths

The following table lists the supported upgrade paths for UDR:

Table 4-1 Supported Upgrade Paths

Source Release	Target Release
24.2.x	24.3.0
24.1.x	24.3.0

4.2 Preupgrade Tasks

This section provides information about preupgrade tasks to be performed before upgrading UDR:

- Take a backup of the current custom-values.yaml file.
- Update the new custom_values.yaml file. For more details about Helm configurable parameters, see UDR Configuration Parameters.
- Provisioning traffic must be stopped when the upgrade procedure is performed.
- Make sure that the current deployment is good for an upgrade before you perform an upgrade on the existing deployment. To verify the current deployment, see Verifying Installation.
- Before upgrading, perform sanity check using Helm test. See Performing Helm Test section for the Helm test procedure.
- Install or upgrade the network policies, if applicable. For more information, see Configuring Network Policies.

4.3 Upgrade Task

This section provides information about the sequence of tasks to be performed for upgrading an existing UDR deployment.

Helm Upgrade

To upgrade UDR 24.2.x to 24.3.0, follow the instructions given below.

> **Note**
>
> During in service upgrade from 24.2.x to 24.3.0, there can be a traffic loss of about 1%.

Upgrading an existing deployment replaces the running containers and pods with new containers and pods. If there is no change in the pod configuration, it is not replaced. Unless there is a change in the service configuration of a microservice, the service endpoints remain unchanged. For example, ClusterIP.

To upgrade multiple site deployment, see Upgrade of Multiple Site Deployments.

Configuring cnDBTier

1. The following step is applicable only for multiple site cnDBTier deployment.

   Update mysqld configuration in the cnDBTier custom values file before the installing or upgrading UDR or SLF.

   ```
   global:
     ndbconfigurations:
       api
         auto_increment_increment: 3
         auto_increment_offset: 1
   ```

   > **Note**
   >
   > - Set the `auto_increment_increment` parameter as same as number of sites. For example: If the number of sites is 2, set its value as 2 and if the number of sites is 3, set its value as 3.
   > - Set the `auto_increment_offset` parameter as site ID. For example: The site ID for Site 1 is 1, Site 2 is 2, for Site 3 is 3 and so on.
   > - If the fresh installation or upgrade of UDR or SLF on cnDBTier is not planned, then run the following command to edit the exiting mysqldconfig configmap on all the cnDBTier sites.
   >
   >   ```
   >   kubectl edit configmap mysqldconfig <db-site-namespace>
   >   ```
   >
   >   **For Example:** `kubectl edit configmap mysqldconfig -n site1`
   >
   >   > **Note**
   >   >
   >   > Update the `auto_increment_increment` and `auto_increment_offset` values as mentioned in the previous step for all sites.

2. The following changes are made in the cnDBTier custom values shared as part of UDR custom templates:
   - The value of `HeartbeatIntervalDbDb` parameter is changed in the cnDBTier custom values files. Before performing Helm upgrade of cnDBTier, you must refer to *Oracle*

Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide for upgrade instructions to perform the configuration changes.

- The disk size is updated and SLF and EIR resource profiles are changed due to the subscriber base increase. You must refer to *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to perform the configuration changes.

Upgrading UDR

⚠ Caution

- Stop the provisioning traffic before you start the upgrade procedure.
- Do not configure UDR while the upgrade is in progress.
- Do not exit the `helm upgrade` command manually. After running the `helm upgrade` command, it takes sometime (depending upon number of pods to upgrade) to upgrade all of the services. Do not press "ctrl+c" to exit the `helm upgrade` command. It may lead to anomalous behavior.

1. Untar the latest OCUDR software components and if required, re-tag and push the images to customer's repository. For more information about extracting OCUDR software components, see Pushing the Images to Customer Docker Registry.
2. Take a backup of the 24.2.x version's <ocudr-values.yaml> file.
3. Modify the ocudr-custom-values-24.3.0 yaml file parameters as per site requirement.
4. Retain the **nfInstanceId** configuration for the site same as OCUDR 24.1.x values. In case of multisite deployments, configure **nfInstanceId** differently for each SLF or UDR.
5. Ensure the **nfInstanceId** configuration in the global section is same as that in the appProfile section of NRF client.

   ⓘ Note

   This step is applicable to all UDR installations from 1.14.x onwards.

6. The following steps are required only when the Control Shutdown feature need to be enabled or it is enabled during upgrade:
   - Perform the following steps when Control Shutdown feature is enabled and upgraded from 23.2.x to 24.3.0:
     a. Log into one of the cnDBtier pods using the `exec` command.
        Example:

        ```
        kubectl exec -it <pod-name> -n <namespace> bash
        ```

     b. Run the following command to choose the database:

        ```
        use configDB
        ```

c. Run the following command to update the supported configuration item for operational state:

```
UPDATE configuration_item SET
cfg_value='{"operationalState":"NORMAL","operationalStateHistory":
[]}' where cfg_key='OPERATIONAL-STATUS';
```

7. The following steps are required only when the Network Policy and Conflict Resolution features need to be enabled during upgrade:
   - Enable Network Policy when upgrading to 24.3.0, see Configuring Network Policies and Network Policy section in the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.
   - Enable Conflict Resolution when upgrading to 24.3.0, see Conflict Resolution section in the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.
8. Assign appropriate values to core_service in the appInfo configuration based on udrMode.

> **Note**
>
> This step is applicable to all UDR installations from 1.14.x onwards.

9. Change the configurations in the ocudr-custom-values-24.3.0.yaml file as per site requirements. For more information about UDR configuration, see UDR Configuration Parameters.
10. Run the following command to upgrade an existing UDR deployment.

```
$ helm upgrade <release> <helm chart> [--version <OCUDR version>] -f
<ocudr-custom-values-24.3.0.yaml>
```

`<release>` is available in the output of `helm list` command.

`<helm chart>` is the name of the chart in the form of <repository/ocudr> e.g. reg-1/ocudr or cne-repo/ocudr.

```
$ helm upgrade <release> -n <namespace> <helm chart> [--version <OCUDR
version>] -f <ocudr-custom-values-24.3.0.yaml>
```

`<release>` is available in the output of `helm list` command.

`<helm chart>` is the name of the chart in the form of <repository/ocudr> e.g. reg-1/ocudr or cne-repo/ocudr.

11. Perform sanity check using Helm test. See Performing Helm Test section for the Helm test procedure.

The below steps are applicable only when there is a helm upgrade failure:

- If the helm upgrade failure is intermittent, check the below error scenarios.

1. If the helm upgrade fails with `<helm-release-name>-nudr-config-server-pre-rollback/<helm-release-name>-nudr-config-server-post-rollback hook running into Error state`, check the logs of the failed hook.
2. Perform Step 3, if the logs show any of the following errors:
   a. "message":"Unknown column 'tb.snapshot_id' in 'where clause'"
   b. "message":"Rolling back to the savepoint and exiting...","thrown":{"commonElementCount":0,"localizedMessage":"Table 'udrconfigdb.snapshot_mapping' doesn't exist","message":"Table udrconfigdb.snapshot_mapping' doesn't exist"
3. Update the database tables using the following workaround SQL statements:
   a. Run the following command to log in to ndbappsql node pods.

      ```
      "kubectl exec <sqlnode> -n <ns> -it bash"
      ```

   b. Copy the below command in codeblock to a commands.sql file. The commands.sql file can be created using vim editor that is present inside mysql pod.
      SQL Commands

      ```
      SET default_storage_engine=NDBCLUSTER;
      use udrconfigdb;
      drop table if exists snapshot_mapping;
      CREATE TABLE IF NOT EXISTS  snapshot_mapping
                  ( snapshot_id bigint(20) NOT NULL AUTO_INCREMENT,
                    `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
                    `created_time` datetime DEFAULT CURRENT_TIMESTAMP,
                   `restored_time` datetime DEFAULT NULL,
                    PRIMARY KEY (`snapshot_id`),
                    UNIQUE KEY `name` (`name`) )
                    AUTO_INCREMENT=21 DEFAULT CHARSET=utf8
      COLLATE=utf8_unicode_ci;
      alter table topic_info_backup add column snapshot_id bigint;
      alter table topic_info_backup add foreign key (snapshot_id)
      references snapshot_mapping(snapshot_id);
      alter table configuration_item_backup add column snapshot_id bigint;
      alter table configuration_item_backup add foreign key (snapshot_id)
      references snapshot_mapping(snapshot_id);
      use udrdb;
      delete from ReleaseConfig where SiteId="5a7bd676-ceeb-44bb-95e0-
      f6a55a328b03" and CfgKey="nrf-client-nfmanagement";
      ```

> **Note**
>
> - The second SQL command `use udrconfigdb` mentioned above must refer to the configuration database name of UDR or SLF
> - The second last SQL command `use udrdb` mentioned above must refer to the subscriber database name of UDR or SLF.
> - The last SQL command `SiteId` mentioned above must be same as the one configured in `global.nfInstanceId` in the ocudr 24.3.0 custom-values.yaml file.

c. Run the below command to load the SQL commands to the database.

```
mysql -h127.0.0.1 -u<username> -p<password> < commands.sql
```

- After performing the above workaround, retry the rollback again using the helm upgrade command.

Upgrade of Multiple Site Deployments

You must follow a specific order during upgrade of multiple site deployment. You must upgrade one site at a time starting with the NF and then cnDBTier. For example, in a three site setup, you must follow the below order for NF and cnDBTier upgrade:

- Upgrade Site 1 UDR, Provisioning Gateway, CNC Console, and Tools (subscriber bulk import and subscriber export tool) deployments if present.
- Upgrade Site 1 cnDBTier
- Upgrade Site 2 UDR, Provisioning Gateway, CNC Console, and Tools (subscriber bulk import and subscriber export tool) deployments if present.
- Upgrade Site 2 cnDBTier
- Upgrade Site 3 UDR, Provisioning Gateway, CNC Console, and Tools (subscriber bulk import and subscriber export tool) deployments if present.
- Upgrade Site 3 cnDBTier

The above upgrade order is required because of the SQL rollback limitation applicable for cnDBTier:

- If there are schema changes on any of the tables when running on the higher n+1 release version of cnDBTier, then when you rollback the NF to "n" version of cnDBtier, the schema changes which happened on the higher n+1 version will not be handled and the tables with these changes are dropped and are not re-created on the lower version. For more information, see the **Prerequisites** section in the *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
- Rollback of cnDBTier must be performed carefully as per the specified order mentioned in Rollback of Multiple Site Deployments because UDR, Provisioning Gateway, and Tools (subscriber bulk import and subscriber export tool) upgrade and rollback procedure includes database schema changes.
- There must not be database schema change in the current cnDBTier version to rollback. If there is any database schema change after cnDBtier is upgraded, then these schema changes must be reverted. You must contact My Oracle Support for assistance.

> **Note**
>
> - To enable conflict resolution feature during upgrade. For more information, see "Conflict Resolution" section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*. You must perform the schema changes before upgrading UDR components and cnDBTier.
> - During the upgrade procedure if there is an issue on UDR, Provisioning Gateway, or Tools in any sites, then UDR, Provisioning Gateway, or Tools must be rolled back immediately. Do not rollback cnDBTier. cnDBTier must be rolled back only if there is a cnDBTier upgrade issues.
> - Verify the cnDBTier after the upgrade is complete. If there are issues, rollback the cnDBTier immediately. Do not rollback UDR, Provisioning Gateway, or Tools, they can remain on the higher version and are compatible with n-1 cnDBTier release.
> - To avoid failures during cnDBTier rollback due to schema changes (for example, helm upgrade, rollback performed on UDR, Provisioning Gateway, and subscriber bulk import and subscriber export tool deployments), you can drop the modified schema tables manually before rollback and re-create the modified schema tables after the rollback is complete. For more information, see the **Prerequisites** section in the *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* and contact My Oracle Support for assistance.

If the Provisioning Gateway, nudr-bulk-import, and nudr-export-tool are deployed in the site, the upgrade must be preformed in the order as follows:

- Upgrade UDR
- Upgrade Provisioning Gateway
- Upgrade nudr-bulk-import
- Upgrade nudr-export-tool
- Upgrade cnDBTier

If the upgrade fails, see the **Debugging Upgrade or Rollback Failure** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.

4.4 Postupgrade Tasks

Perform the following steps if your upgrading from 24.1.x to 24.3.0 release.

> **Note**
>
> These steps are not applicable if your upgrading from 24.2.x to 24.3.0 release.

From 24.1.0 release onwards the following features are supported using REST API. The features are migrated from HELM to REST API mode during upgrade procedure. After upgrading from 24.1.x to 24.3.0 release version the following features are disabled by default:

- Server Header Support on IGW-Sig
- User Agent Header Support on IGW-Sig
- User Agent Header Support on EGW

Perform the following steps post the Helm upgrade to configure the following features:

- **Server Header Support on IGW-Sig**:
  - Perform a GET operation on `http://<nudr_config_svc_IP>:<port>/udr/nf-common-component/v1/igw-sig/serverheaderdetails` to get the current configurations of the feature.

```
{
    "enabled": false,
    "errorCodeSeriesId": "E1",
    "configuration": {
        "nfType": "",
        "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03"
    }
}
```

  - Perform a PUT operation on `http://<nudr_config_svc_IP>:<port>/udr/nf-common-component/v1/igw-sig/serverheaderdetails` to enable the feature by setting the `enabled` parameter to true and configuring the `nfType` and `errorCodeSeriesId` as required.

```
{
    "enabled": true,
    "errorCodeSeriesId": "E1",
    "configuration": {
        "nfType": "UDR",
        "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03"
    }
}
```

  - Perform a PUT operation on `http://<nudr_config_svc_IP>:<port>/udr/nf-common-component/v1/igw-sig/errorcodeserieslist` to configure the error code series list.

```
[
    {
        "id": "P1",
        "exceptionList": [
            "RequestTimeout",
            "ConnectionTimeout",
            "UnknownHostException",
            "NotFoundException"

        ],
        "errorCodeSeries":
        [
            {
                "errorSet": "5xx",
                "errorCodes": [503]
            }
        ]
    },
    {
        "id": "E1",
        "errorCodeSeries": [
```

```
            {
                "errorSet": "4xx",
                "errorCodes": [
                    -1
                ]
            },
            {
                "errorSet": "5xx",
                "errorCodes": [
                    -1
                ]
            }
        ]
    }
]
```

- **User Agent Support on IGW-Sig**:
  - Perform a GET operation on `http://<nudr_config_svc_IP>:<port>/udr/nf-common-component/v1/igw-sig/useragentheadervalidation` to get the current configurations of the feature.

    ```
    {
        "enabled": false,
        "validationType": "strict",
        "consumerNfTypes": [
            "PCF",
            "NRF",
            "UDM"
        ]
    }
    ```

  - Perform a PUT operation on `http://<nudr_config_svc_IP>:<port>/udr/nf-common-component/v1/igw-sig/useragentheadervalidation` to enable the feature by setting the `enabled` parameter to true.

    ```
    {
        "enabled": true,
        "validationType": "strict",
        "consumerNfTypes": [
            "PCF",
            "NRF",
            "UDM"
        ]
    }
    ```

- **User Agent Support on EGW**:
  - Perform a GET operation on `http://<nudr_config_svc_IP>:<port>/udr/nf-common-component/v1/egw/useragentheader` to get the current configurations of the feature.

    ```
    {
        "nfFqdn": "",
        "nfType": "",
        "enabled": false,
    ```

```
    "nfInstanceId": "",
    "addFqdnToHeader": false,
    "overwriteHeader": false
}
```

- Perform a PUT operation on `http://<nudr_config_svc_IP>:<port>/udr/nf-common-component/v1/egw/useragentheader` to enable the feature by setting the `enabled` parameter to true and configuring the mandatory. `nfType` parameter as required.

  ```
  {
      "nfFqdn": "",
      "nfType": "UDR",
      "enabled": true,
      "nfInstanceId": "",
      "addFqdnToHeader": false,
      "overwriteHeader": false
  }
  ```

For more information about REST API, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

5
Rolling Back UDR

This chapter provides information about rolling back Oracle Communications Cloud Native Core, Unified Data Repository (UDR) deployment to the previous release.

5.1 Supported Rollback Paths

The following table lists the supported rollback paths for UDR:

Table 5-1 Supported Rollback Paths

Source Release	Target Release
24.3.0	24.2.x
24.3.0	24.1.x

To rollback to the previous UDR version, run the following command to check if the pods have successfully started (shows running state).

```
kubectl get pods -n <namespace_name>
```

Where, `<namespace_name>` is the UDR namespace name. For example, `ocudr`.

If the pod logs or describe pods command results are not sufficient to handle the issue, then rollback using Helm and proceed with Rollback DB Schema instructions.

5.2 Rollback Tasks

Run the following command to check if the pods are successfully started:

```
kubectl get pods -n <namespace_name>
```

> **Note**
>
> You have to rollback if there are issues that cannot be recovered by checking the logs. You must perform a helm rollback first and then proceed with rollback database schema instructions.

Use the backup of the 24.2.x version's <ocudr-values.yaml> file to roll back from UDR 24.3.0 to a previous version:

1. Run the following command to obtain the revision number:

   ```
   helm history <helm release name>
   ```

   Example: `helm history ocudr`

2. Run the following rollback command.

   > **Note**
   >
   > For multisite deployments, run the rollback command for all sites.

   ```
   helm rollback <helm release name> <revision_no>
   ```

   Where, `<helm release_name>` is the release name used by the Helm command.

   `<revision_no>` is the release number to which UDR needs to be rolled back.

   Example: `helm rollback ocudr 1`

3. Run the following command when you upgrade from 24.1.x or 24.2.x to 24.3.0 and then rollback to 24.1.x or 24.2.x:

   ```
   Login to mysql, "kubectl exec <dataNodes> -n <namespace> -it bash"
   Login to mysql password: mysql -h 127.0.0.1 -uroot -pNextGenCne
   use <udrdb-name>;
   Example: use udrdb;
       delete from ReleaseConfig where SiteId=<siteId> and CfgKey=ocudr;
   ```

Rollback of Multiple Site Deployments

You must follow a specific order during rollback of multiple site deployment. You must rollback all cnDBTier sites and then rollback UDR across all sites. For example, in a three site setup you must follow the below order for NF and cnDBTier rollback:

- Rollback site 1 cnDBTier
- Rollback site 2 cnDBTier
- Rollback site 3 cnDBTier
- Rollback Site 1 UDR, Provisioning Gateway, CNC Console, and reinstall the rollback version of the tools (subscriber bulk import and subscriber export tool) deployments if present.
- Rollback Site 2 UDR, Provisioning Gateway, CNC Console, and reinstall the rollback version of the tools (subscriber bulk import and subscriber export tool) deployments if present.
- Rollback Site 3 UDR, Provisioning Gateway, CNC Console, and reinstall the rollback version of the tools (subscriber bulk import and subscriber export tool) deployments if present.

The above rollback order is required because of the SQL rollback limitation applicable for cnDBTier:

- If there are schema changes on any of the tables when running on the higher n+1 release version of cnDBTier, then when you rollback the NF to "n" version of cnDBtier, the schema changes which happened on the higher n+1 version will not be handled and the tables with these changes are dropped and are not re-created on the lower version. For more information, see the **Prerequisites** section in the *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

- Rollback of cnDBTier must be performed carefully as per the above specified order because UDR, Provisioning Gateway, and Tools (subscriber bulk import and subscriber export tool) upgrade and rollback procedure includes database schema change.
- There must not be database schema change in the current cnDBTier version to rollback. If there is any database schema change after cnDBtier is upgraded, then these schema changes must be reverted. You must contact My Oracle Support for assistance.

> **Note**
>
> - You must disable the conflict resolution feature during rollback. For more information, see "Conflict Resolution" section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*. You must perform the schema changes before upgrading UDR components and cnDBTier. Schema changes performed during upgrade can be retained since the changes do not impact when you rollback the UDR version.
> - During the upgrade procedure if there is an issue on UDR, Provisioning Gateway, or Tools in any sites, then UDR, Provisioning Gateway, or Tools must be rolled back immediately. Do not rollback cnDBTier. cnDBTier must be rolled back only if there is a cnDBTier upgrade issues.
> - Verify the cnDBTier after the upgrade is complete. If there are issues, rollback the cnDBTier immediately. Do not rollback UDR, Provisioning Gateway, or Tools, they can remain on the higher version and are compatible with n-1 cnDBTier release.
> - To avoid failures during cnDBTier rollback due to schema changes (for example, helm upgrade, rollback performed on UDR, Provisioning Gateway, and subscriber bulk import and subscriber export tool deployments), you can drop the modified schema tables manually before rollback and re-create the modified schema tables after the rollback is complete. For more information, see the **Prerequisites** section in the *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* and contact My Oracle Support for assistance.

If the rollback fails, see the **Debugging Upgrade or Rollback Failure** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.

5.3 Postrollback Tasks

After performing rollback, restore preupgrade data obtained earlier through manual backup.

GET API for different resources can be used to see current values in UDR, then accordingly if any update required, individual service APIs defined for different resources can be used to reconfigure the data backup taken before upgrade. See *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide for API details.*

6
Uninstalling UDR

This chapter provides information about uninstalling Oracle Communications Cloud Native Core, Unified Data Repository (UDR).

6.1 Uninstalling UDR Using Helm

To uninstall or completely delete the UDR deployment, run the following command:

```
helm uninstall <helm_release_name_for_ocudr> --namespace <ocudr_namespace>
```

Where,

`<helm-release>`

is a name provided to identify the Helm deployment.

`<namespace>`

is the name provided to identify the namespace of UDR deployment.

Example:

```
helm uninstall ocudr -n ocudr
```

> **Important**
>
> As part of UDR uninstallation, its databases are not dropped. If you want to drop the databases, see Removing UDR Database.

> **Note**
>
> The `kubectl` commands may vary depending on the platform deployment. Replace kubectl with Kubernetes environment-specific command line tool to configure Kubernetes resources through kube-api server. The instructions provided in this section are as per the Oracle Communications Cloud Native Environment (OCCNE) version of kube-api server.

> **⚠ Caution**
>
> User, computer and applications, and character encoding settings may cause an issue when copy-pasting commands or any content from PDF. The PDF reader version also affects the copy-pasting functionality. It is recommended to verify the pasted content especially when the hyphens or any special characters are part of the copied content.

Helm keeps a record of its releases, so you can still reactivate the release after uninstalling it.

To completely remove a release from the cluster, add the `--purge` parameter to `helm delete` command:

```
helm delete --purge release_name
```

For example:

```
helm delete --purge ocudr
```

To verify the UDR uninstallation, run the following command:

```
kubectl get all -n <release-namespace>
```

In case of successful uninstallation, no UDR resource is displayed in the command output.

If the command output displays the UDR resources or objects, then perform Deleting Kubernetes Namespace.

If the command output displays any Kubernetes namespace, then perform Deleting Kubernetes Namespace.

6.2 Deleting Kubernetes Namespace

This section describes how to delete Kubernetes namespace where UDR is deployed.

Run the following command to delete the Kubernetes namespace:

```
kubectl delete namespace <release_namespace>
```

Where,

`<release_namespace>` is the deployment namespace used by the Helm command.

For example:

```
kubectl delete namespace ocudr
```

6.3 Removing UDR Database

This section describes how to remove database when uninstalling UDR.

Single Site Deployment

Run the following command to remove UDR subscriber and configurations database on single site setup

> **Note**
>
> When dropping <udrdb_name> or <udrconfigdb_name> database, make sure that there are no service or tools like nudr-bulk-import or nudr-odmand-migration running. If the service or tools are running, you must uninstall UDR deployment and UDR tools.

1. Run the following command to log in to one of the ndb app SQL node pods.

   ```
   kubectl exec -it ndbappmysqld-0 -n <dbtier-ns> bash
   ```

   Example:

   ```
   kubectl exec -it ndbappmysqld-0 -n occne-rs1 bash
   ```

2. Run the following command to log in to the SQL terminal using UDR database user credentials.

   ```
   mysql -h127.0.0.1 -u<username> -p<password>
   ```

   Example:

   ```
   mysql -h 127.0.0.1 -uudruser -pudrpasswd
   ```

3. Run the following commands to drop the configuration and subscriber databases.

   ```
   drop database <udrdb_name>;
   drop database <udrconfigdb_name>;
   ```

   Example:

   ```
   drop database udrdb;
   drop database udrconfigdb;
   ```

Multiple Site Deployment

Run the following command to remove UDR subscriber and configurations databases on multisite setup.

> **NOT_SUPPORTED**
>
> Perform the following steps on one of the sites by logging into one of the ndb app SQL nodes.

> **Note**
>
> When dropping <udrdb_name> or <udrconfigdb_name> DB for multiple site , Make sure there are no service or tools like nudr-bulk-import or nudr-odmand-migration running in any site. If the service or tools are running, you must uninstall UDR deployment and UDR tools from all site.

1. Run the following command to log in to one of the ndb app SQL node pods on one of the sites.

   ```
   kubectl exec -it ndbappmysqld-0 -n <dbtier-ns> bash
   ```

   Example:

   ```
   kubectl exec -it ndbappmysqld-0 -n occne-rs1 bash
   ```

2. Run the following command to log in to the SQL terminal using UDR database user credentials.

   ```
   mysql -h127.0.0.1 -u<username> -p<password>
   ```

   Example:

   ```
   mysql -h 127.0.0.1 -uudruser -pudrpasswd
   ```

3. Run the following commands to drop the configuration and subscriber databases.

   ```
   drop database <udrdb_name>;
   drop database <udrconfigdb_name>;
   ```

   Example:

   ```
   drop database udrdb;
   drop database udrconfigsite1db;
   drop database udrconfigsite2db;
   drop database udrconfigsite3db;
   ```

6.4 Cleaning UDR Deployment

This section describes how to clean the UDR deployment:

1. Run the following command to remove all failed Helm releases:

   ```
   helm ls --all
   ```

2. If UDR Helm release is in a failed state, then run the following command to uninstall the namespace:

   ```
   helm3 uninstall <release-name> -n <release-namespace>
   ```

> **Note**
>
> If this is taking more time as it is running the delete hook jobs, then run the following script in parallel (in another session) to clear all the delete jobs.
>
> ```
> while true; do kubectl delete jobs --all -n <release-namespace>;
> sleep 5;done
> ```

Monitor the "helm delete <release-namespace>" command. Once that is succeeded, press "ctrl+c" to stop the above command.

3. Run the following command to clean all the Kubernetes objects:

   ```
   kubectl get all -n <release-namespace>
   ```

   It gives a detailed overview of the current objects of <release-namespace>. Delete all those objects.

   > **Caution**
   >
   > Be sure before running the following commands as it deletes all objects of Kubernetes in the specified namespace. In case the user has created RBAC and service account details before Helm install in the same namespace and is required, then do not delete them. In case, custom service account was not provided by the user and it is safe to remove it, then run the following commands to delete the resources or objects which are not required:
   > Deleting all the Kubernetes objects:
   >
   > ```
   > kubectl delete all --all -n <release-namespace>
   > ```
   >
   > Deleting all the current configmaps:
   >
   > ```
   > kubectl delete cm --all -n <release-namespace>
   > ```

4. Run the following command to view all the pending resources not cleaned during helm uninstall.

   ```
   kubectl get all -n <release-namespace>
   ```

5. Run the following command to delete the pending resources:

   ```
   kubectl delete <resource-type> <resource-name> -n <release-namespace>
   ```

6. Run the following command to delete Kubernetes namespace:

> ⚠ **Caution**
>
> Be sure before removing the namespace. It deletes all the resources or objects created in the namespace.

```
kubectl delete namespace <ocudr kubernetes namespace>
```

Example:

```
kubectl delete namespace ocudr
```

7
Fault Recovery

This chapter describes the procedures to perform fault recovery for Oracle Communications Cloud Native Core, Unified Data Repository (UDR) deployment.

7.1 Overview

The UDR operators can take database backup and restore them either on the same or a different cluster. The UDR database (MySQL NDB Cluster) is used for running any command or to follow any instruction.

> **Note**
>
> This guide describes recovery procedures to restore UDR database only. To restore all the databases that are part of cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide.* available on My Oracle Support (MOS).

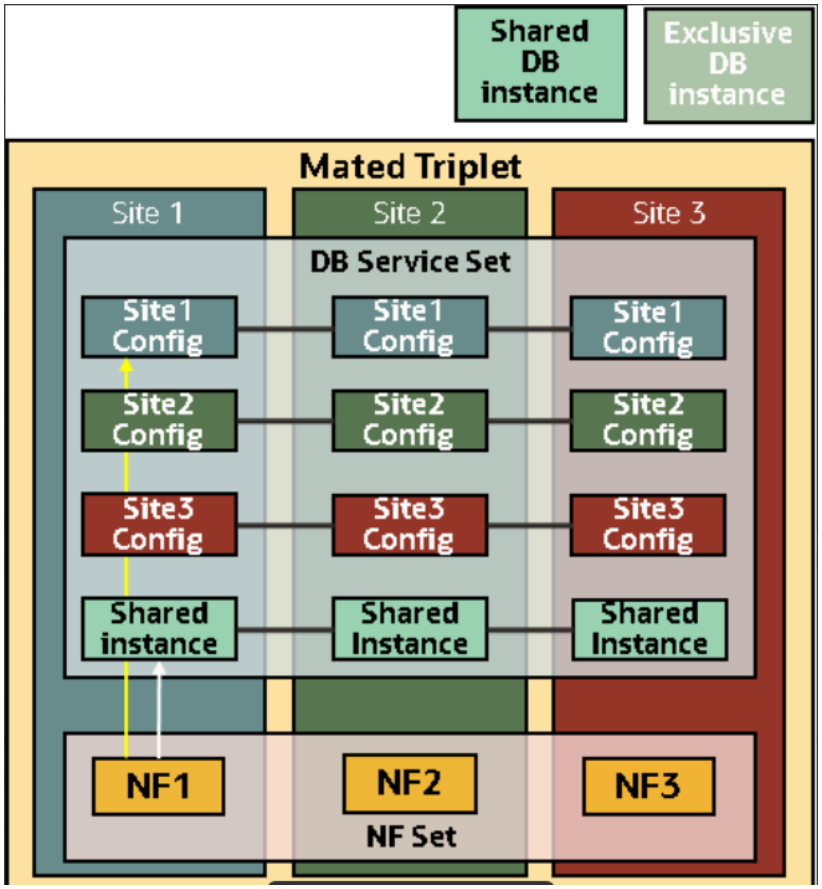
Database Model of UDR

UDR database consists of configuration data and subscribers data. The details are as follows:

- Configuration Data: The configuration data is exclusive for a given site. Thus, an exclusive logical database is created and used by a site to store its configuration data. Using CNC Console and Configuration Management service, an operator can configure data in the respective site only.
- Subscribers Data: The subscribers data is shared across sites. Thus, a common logical database is created and used by all sites. The data is replicated across sites to preserve and share session with mated sites. In case of cross-sites messaging or a site failure, shared session data helps in continuity of service.

The following image shows UDR database model in three different sites.

Figure 7-1 Database Model



7.2 Impacted Areas

The following table provides information about the impacted areas during UDR fault recovery:

Table 7-1 Impacted Areas

Scenario	Requires Fault Recovery or re-install of CNE?	Requires Fault Recovery or Re-install of cnDBTier?	Requires Fault Recovery or re-install of UDR?	Other
Scenario 1: Database Migration to a New Cluster	Yes, if cluster is not present	Yes, if cnDBTier is not present	Yes	
Scenario 2: Deployment Failure	No	No	Yes	
Scenario 3: Database Corruption				

Table 7-1 (Cont.) Impacted Areas

Scenario	Requires Fault Recovery or re-install of CNE?	Requires Fault Recovery or Re-install of cnDBTier?	Requires Fault Recovery or re-install of UDR?	Other
Scenario 3A: Configuration Database Corruption	No	No	No	Configuration database is present in the site exclusive database. UDR backup and restore of configuration database is required on impacted site.
Scenario 3B: Subscriber Database Corruption	No	Yes Restore DBTier from older backup is the only way to restore back to restore point.	No Only if cnDBTier credentials are changed.	All sites require Fault Recovery.
Scenario 4: Site Failure	Yes	Yes	Yes	
Manual Database Backup and Restore	No	No	Yes	

7.3 Prerequisites

Before you run any Fault recovery procedure, ensure that the following prerequisites are met:

- DBTier should be in healthy state and available on multiple sites along with UDR
- Automatic backup should be enabled on DBTier. Scheduled regular backups help to:
  - Restore stable version of the UDR database
  - Minimize significant loss of data due to upgrades or roll back failures
  - Minimize loss of data due to system failure
  - Minimize loss of data due to data corruption or deletion due to external input
  - Migrate UDR database information from one site to another
- Custom values file used at the time of UDR deployment is retained

7.4 Fault Recovery Scenarios

This section describes the fault recovery procedures for various scenarios.

7.4.1 Scenario 1: Database Migration to a New Cluster

This section describes how to migrate UDR from an existing cluster to a new cluster. Some of the reasons for migrating UDR are as follows:

- Database failure
- Moving UDR to a bigger cnDBTier
- Moving UDR to a new (freshly installed) cnDBTier where upgrade is not supported

This scenario includes the following sub scenarios.

7.4.1.1 Scenario 1A: Configuration Database Migration

A scenario where you want to migrate only the configuration database to a new cluster.

To migrate configuration database:

1. Export configuration data from an older site to a file. For information about exporting data, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.
2. (Optional) Install Kubernetes cluster on a new site. For information about installing Kubernetes cluster, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.
3. (Optional) Install cnDBTier on new site. For information about installing cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
4. Install UDR. For more information about UDR installation, see Installing UDR Package.
5. Import the configuration data from a file to which you have exported configuration data in Step1. For more information about importing data, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

> **Note**
>
> Common services does not provide export and import of their configuration data. This requires rerun of its configuration APIs on a new site to reconfigure common services.

7.4.1.2 Scenario 1B: Configuration and Subscriber Database Migration

A scenario where you want to migrate the configuration as well as subscriber database to a new cluster.

To migrate the configuration and subscriber database to a new cluster:

1. Shutdown an older site. For information about shutting down a site, see Uninstalling UDR.
2. (Optional) Install Kubernetes cluster on a new site. For information about installing Kubernetes cluster, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.
3. (Optional) To take data backup from an older site and restore it to a new site, perform DBTier Fault recovery procedure. For more information about DBTier backup, see the **Create On-demand Database Backup** chapter and to restore database to a new site, see the **Restore DB with Backup** chapter in the *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
4. Install UDR. For more information, see Installing UDR Package.

> **Note**
>
> This procedure migrates both, the configuration as well as subscriber data to a new cluster.

7.4.2 Scenario 2: Deployment Failure

This scenario describes how to recover UDR when its deployment corrupts.

To recover UDR:

1. Uninstall UDR. For more information, see Uninstalling UDR.
2. Install UDR. For more information, see Installing UDR Package.

7.4.3 Scenario 3: Database Corruption

This scenario describes how to recover UDR when its database (configuration database and subscriber database) corrupts.

7.4.3.1 Scenario 3A: Configuration Database Corruption

This section describes how to recover UDR configuration database from the corrupted database. The configuration database is stored in a site exclusive database along with its tables. Thus, corruption of configuration database impacts only a particular site.

To recover UDR configuration database:

1. Export the configuration data using either Cloud Native Core Console or RESTAPI. For more information about exporting configuration database, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.
2. Import the configuration data on new site. For more information about importing data, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

> **Note**
>
> Common services does not provide export and import of their configuration data. This requires rerun of its configuration APIs on a new site to reconfigure common components.

7.4.3.2 Scenario 3B: Subscriber Database Corruption

This section describes how to recover UDR when its subscriber database corrupts.

When the subscriber database corrupts, the database on all other sites may also corrupt due to data replication. It depends on the replication status after the corruption has occurred. If the data replication is broken due to database corruption, then DBTier fails in either single or multiple sites (not all sites). And if the data replication is successful, then database corruption replicates to all the DBTier sites and DBTier fails in all sites.

The fault recovery procedure covers following sub-scenarios:

7.4.3.2.1 When cnDBTier Failed in Single or Multiple Sites

This section describes how to recover subscriber database when the data replication is broken due to database corruption and DBTier has failed in either single or multiple sites (not all sites).

To recover subscriber database:

1. Uninstall UDR. For information, see Uninstalling UDR.
2. For cnDBTier fault recovery:
   a. To recover a single node failure in cnDBTier, follow the **Restoring Single Node Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
   b. If a stand alone cnDBTier is down, follow the **Restoring Database From Backup** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the cluster.
   c. If there is georeplication failure between multiple cnDBTier clusters in replication, follow the **Restoring Georeplication Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the data.

   > **Note**
   >
   > The Restore Georeplication Failure chapter has a procedure for two sites where one of the cluster has fatal error. You can perform that procedure for all the sites in a multiple site setup.

3. Run the below curl command to check whether the site replication is UP or not, once the fault recovery procedure completed:

   ```
   (curl http://mysql-cluster-db-monitor-svc.occne-cndbtier:8080/db-tier/
   status/replication/realtime)
   ```

4. Run the below curl command on both the sites to check grstatus is COMPLETED or not:

   ```
   curl -X GET http://<svcname of replication svc>.<ns>/db-tier/gr-recovery/
   site/cluster1/status


   curl -X GET http://<svcname of replication svc>.<ns>/db-tier/gr-recovery/
   site/cluster2/status
   ```

5. Install UDR. For more information, see Installing UDR Package.

7.4.3.2.2 When cnDBTier Failed in All Sites

This section describes how to recover subscriber database when successful data replication corrupts all the DBTier sites.

To recover subscriber database:

1. Uninstall UDR on all sites. For more information. see Uninstalling UDR.
2. For cnDBTier fault recovery:

a. To recover a single node failure in cnDBTier, follow the **Restoring Single Node Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

b. If a stand alone cnDBTier is down, follow the **Restoring Database From Backup** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the cluster.

c. If there is georeplication failure between multiple cnDBTier clusters in replication, follow the **Restoring Georeplication Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the data.

> **Note**
>
> The Restore Georeplication Failure chapter has a procedure for two sites where one of the cluster has fatal error. You can perform that procedure for all the sites in a multi-site setup.

3. Run the below curl command to check whether the site replication is UP or not, once the fault recovery procedure completed:

```
(curl http://mysql-cluster-db-monitor-svc.occne-cndbtier:8080/db-tier/
status/replication/realtime)
```

4. Run the below curl command on both the sites to check grstatus is COMPLETED or not:

```
curl -X GET http://<svcname of replication svc>.<ns>/db-tier/gr-recovery/
site/cluster1/status


curl -X GET http://<svcname of replication svc>.<ns>/db-tier/gr-recovery/
site/cluster2/status
```

5. Install UDR. For more information, see Installing UDR Package.

7.4.4 Scenario 4: Site Failure

This section describes how to perform fault recovery when either one, multiple, or all sites have a software failure. The following are site failure scenarios:

7.4.4.1 Scenario 4A: Single or Multiple Site Failure

This scenario applies when one or more sites, and not all sites, have failed and there is a requirement to perform fault recovery. It is assumed that the user has cnDBTier and Oracle Communications UDR installed on multiple sites with automatic data replication and backup enabled.

To recover the failed sites:

1. Run the Cloud Native Environment (CNE) installation procedure to install a new cluster. For more information, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.
2. For cnDBTier fault recovery:

a. To recover a single node failure in cnDBTier, follow the **Restoring Single Node Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

b. If a stand alone cnDBTier is down, follow the **Restoring Database From Backup** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the cluster.

c. If there is georeplication failure between multiple cnDBTier clusters in replication, follow the **Restoring Georeplication Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the data.

3. Run the below curl command to check whether the site replication is UP or not, once the fault recovery procedure completed:

```
curl http://mysql-cluster-db-monitor-svc.occne-cndbtier:8080/db-tier/
status/replication/realtime
```

4. Run the below curl command on both the sites to check grstatus is COMPLETED or not:

```
curl -X GET http://<svcname of replication svc>.<ns>/db-tier/gr-recovery/
site/cluster1/status
```

```
curl -X GET http://<svcname of replication svc>.<ns>/db-tier/gr-recovery/
site/cluster2/status
```

5. Install UDR. For more information, see Installing UDR Package.

7.4.4.2 Scenario 4B: All Sites Failure

This scenario applies when all sites have failed and there is a requirement to perform fault recovery. It is assumed that the user has DBTier and Oracle Communications UDR installed on multiple sites with automatic data replication and backup enabled.

To recover all the failed sites:

1. Run the Cloud Native Environment (CNE) installation procedure to install a new cluster. For more information, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.
2. For cnDBTier fault recovery:

a. To recover a single node failure in cnDBTier, follow the **Restoring Single Node Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

b. If a stand alone cnDBTier is down, follow the **Restoring Database From Backup** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the cluster.

c. If there is georeplication failure between multiple cnDBTier clusters in replication, follow the **Restoring Georeplication Failure** chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* to restore the data.

> **Note**
>
> - The auto-data backup file is one that built from scheduled automatic backup.
> - The Restore Georeplication Failure chapter contains a procedure for two sites where one of the cluster has fatal error. You can perform the same procedure for all the sites in a multiple site setup.

3. Install UDR. For more information, see Installing UDR Package.

7.5 Manual Database Backup and Restore

This section describes how to backup and restore UDR database manually. It includes the following procedures:

7.5.1 Backing up the UDR Database

The UDR Database has both, configuration data and provisioned subscriber data in a single database. If there are different databases, then perform the following backup procedure per database.

> **Note**
>
> - UDR database has both configuration data and provisioned subscriber data in a single database. If there are multiple databases, then the following backup procedure must be performed per database.
> - The backup procedure may take longer time depending on the size of the database. Hence, restoring it may be stale. The user must consider this before performing these actions.
> - The commands used to take database backup and restore are provided by the MySQL cluster of DBTier.

1. > **Note**
   >
   > This step is OPTIONAL. By running the uninstall command, the system ensures that there are no pending updates on UDR database. Alternatively, stop (disable) the provisioning updates on UDR and redirect the signalling traffic to another UDR database so that this particular UDR does not have newer updates at the time of data backup.

   Run the following command to ensure all the database transactions are committed.

   ```
   helm3 uninstall <deploymentName> -n <Namespace>
   ```

   where,

   `<deploymentName>` is the helm release name given to run helm3 install command.

   `<Namespace>` is the namespace, which is used to create UDR Kubernetes objects. All the UDR microservices are deployed in this Kubernetes namespace.

Example: `helm uninstall ocudr -n myudr`

2. Log in to the respective SQL node (or API node) and run the following command to take backup of UDR database and UDR configuration database.

```
mysqldump --quick -h127.0.0.1 –u <username> -p <udr databasename>| gzip >
<udr data backup_filename>.sql.gz


mysqldump --quick -h127.0.0.1 –u <username> -p <udr configuration
databasename>| gzip > <udr configuration data backup_filename>.sql.gz
```

where,

`<username>` is the MySQL login username.

`<databasename>` is the name of the database that UDR is currently using.

`<backup_filename>` is the user defined backup file name.

Example:

```
mysqldump --quick -h127.0.0.1 -uudruser -p udrdb | gzip >
udrdbBackup.sql.gz
mysqldump --quick -h127.0.0.1 -uudruser -p udrConfigDb | gzip >
udrConfigDbBackup.sql.gz
```

> **Note**
>
> Use the UDR database name in the command and enter the password when prompted.

> **Note**
>
> There should be enough storage space on the current directory to save the backup file. Depending on the subscribers count, the backup file size may be large and it may take some time to take backup.

The backup file, '**<backup_filename>.sql.gz**' gets created that is used to Restoring the UDR Database.

3. Install UDR again (if required). For more information, see Installing UDR Package.

7.5.2 Restoring the UDR Database

To restore the UDR database:

1. On the new database cluster (DBTier), log in to MySQL NDB Cluster's SQL node and create a new UDR database where the backed up database needs to be restored.

> **Note**
>
> For more details about creating the database, see PreInstallation Tasks.

2. Copy the <backup_ filename >.sql.gz file to new UDR database.

> **Note**
>
> In case the back up file corrupts, take the UDR database backup again following the Backing up the UDR Database shared in the previous section.

3. Run the following command to restore the database to the new UDR database and enter the password when it prompts.

```
gunzip < <udrdb backup_filename>.sql.gz | mysql -h127.0.0.1 –u <username> -
p < databaseName >
gunzip < <udr configuration db backup_filename>.sql.gz | mysql -h127.0.0.1
–u <username> -p < databaseName >
```

where, <backup_filename> is the user defined backup file name

<username> is the MySQL login username

< databaseName > is the name of the database that UDR is currently using

Example:

```
gunzip < udrdbBackup.sql.gz | mysql -h127.0.0.1 -uudruser -p newUdrdb
gunzip < udrConfigDbBackup.sql.gz  | mysql -h127.0.0.1 -uudruser -p
newUdrConfigurationDb
```

4. Install UDR and connect it to the newly created database.
5. Verify the sanity of UDR after connecting with the new database. For more information, see Post Installation Task.

A

Creating Private Keys and Certificates for Ingress Gateway

> **⚠ Caution**
>
> Creating keys and certificates are outside UDR scope and user or operator should perform this at their discretion.

Following are the certificate requirements when signed by a Certificate authority (CA):

- Ingress Gateway and Egress Gateway does not support certificates with Distinguished Encoding Rules (DER) binary encoding. You must decode the certificates before creating secret.
- Certificates must be signed with a valid hash algorithm. For example, SHA256. Deprecated or weak algorithms, for example SHA1 will raise exceptions.
- Additional restriction imposed in the certificate must be performed carefully. For example, when configuring certificates with extended key usage, you must consider that Ingress Gateway works as a TLS Web server and Egress Gateway work as a TLS Web client.

Gateway supports both RSA and ECDSA signing for ingress and egress traffic. Initial algorithm selection is configured by using `initialAlgorithm` flag in the custom-values.yaml for each gateway. Select as follow:

- For RSA use RS256
- For ECDSA use ES256

Certificates for selected initialAlgorithm must be included in the ocudr-gateway-secret.

To create the private keys and certificates:

1. Run the following command to generate RSA private key:

   ```
   openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout
   rsa_private_key -out rsa_certificate.crt -config ssl.conf -passin
   pass:"keystorepasswd" -passout pass:"keystorepasswd"
   ```

2. Run the following command to convert the private key to .pem format:

   ```
   openssl rsa -in rsa_private_key -outform PEM -out
   rsa_private_key_pkcs1.pem -passin pass:"keystorepasswd" -passout
   pass:"keystorepasswd"
   ```

3. Run the following command to generate a certificate using private key:

   ```
   openssl req -new -key rsa_private_key -out apigatewayrsa.csr -config
   ssl.conf -passin pass:"keystorepasswd" -passout pass:"keystorepasswd"
   ```

Note

Use the following sample file of ssl.conf to configure default entries along with storage area network (SAN) details for your certificate.

```
#ssl.conf
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
req_extensions = req_ext
[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = IN
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Karnataka
localityName = Locality Name (eg, city)
localityName_default = Bangalore
organizationName = Organization Name (eg, company)
organizationName_default = Oracle
commonName = Common Name (e.g. server FQDN or YOUR name)
commonName_max = 64
commonName_default = localhost
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
IP = 127.0.0.1
DNS.1 = localhost
```

4. Run the following set of commands to create root certificate authority (CA):

```
openssl req -new -keyout cakey.pem -out careq.pem -config ssl.conf -passin
pass:"keystorepasswd" -passout pass:"keystorepasswd"
openssl x509 -signkey cakey.pem -req -days 3650 -in careq.pem -out
caroot.cer -extensions v3_ca -passin pass:"keystorepasswd"
echo 1234 > serial.txt
```

5. Run the following command to sign the server certificate with root CA private key:

```
openssl x509 -CA caroot.cer -CAkey cakey.pem -CAserial serial.txt -req -in
apigatewayrsa.csr -out apigatewayrsa.cer -days 365 -extfile ssl.conf -
extensions req_ext -passin pass:"keystorepasswd"
```

6. Run the following command to generate ECDSA private key:

```
openssl ecparam -genkey -name prime256v1 -noout -out ecdsa_private_key.pem
openssl pkcs8 -topk8 -in ecdsa_private_key.pem -inform pem -out
ecdsa_private_key_pkcs8.pem -outform pem -nocrypt
```

7. Run the following command to generate a certificate using private key:

```
openssl req -new -key ecdsa_private_key_pkcs8.pem -x509 -nodes -days 365 -
out ecdsa_certificate.crt -config ssl.conf
openssl req -new -key ecdsa_private_key_pkcs8.pem -out apigatewayecdsa.csr
```

```
-config ssl.conf -passin pass:"keystorepasswd" -passout
pass:"keystorepasswd"
```

8. Run the following command to sign the server certificate with root CA private key:

```
openssl x509 -CA caroot.cer -CAkey cakey.pem -CAserial serial.txt -req -in
apigatewayecdsa.csr -out apigatewayecdsa.cer -days 365 -extfile ssl.conf -
extensions req_ext -passin pass:"keystorepasswd"
```

9. Create key.txt by entering any password.
   Example: `echo "keystorepasswd" > key.txt`

   This password is used to configure gateway key store.

10. Create trust.txt by entering any password.
    Example: `echo "truststorepasswd" > trust.txt`

    This password is used to configure gateway trust store.

11. Run the following commands to create a secret:

```
kubectl create ns NameSpace
kubectl create secret generic ocudr-gateway-secret --from-
file=apigatewayrsa.cer --from-file=caroot.cer --from-
file=apigatewayecdsa.cer --from-file=rsa_private_key_pkcs1.pem --from-
file=ecdsa_private_key_pkcs8.pem --from-file=key.txt --from-file=trust.txt
-n <Namespace>
```

Updating Keys and Certificates in the Existing Secrets

Prerequsite: The certificates and files that need to be updated must be present in the secret.

Perform the following steps to update the existing certificates in secrets:

1. Run the following command to add a certificate:

```
TLS_CRT=$(base64 < "<certificate-name>" | tr -d '\n')
kubectl patch secret <secret-name> -p "{\"data\":{\"<certificatename>\":\"$
{TLS_CRT}\"}}"
```

   Here,

   <certificate-name> is the certificate file name.

   <secret-name> is the name of the secret, for example, `ocudr-gateway-secret`.

   Example:

   Run the following command to add a Certificate Authority (CA) Root from the `caroot.cer` file to the `ocudr-gateway-secret`:

```
TLS_CRT=$(base64 < "caroot.cer" | tr -d '\n')
kubectl patch secret ocudr-gateway-secret -p "{\"data\":{\"caroot.cer\":\"$
{TLS_CRT}\"}}" -n udr
```

   Similarly, you can also add other certificates and keys to the `ocudr-gateway-secret`.

2. Run the following command to update an existing certificate:

```
TLS_CRT=$(base64 < "<updated-certificate-name>" | tr -d '\n')
kubectl patch secret <secret-name> -p "{\"data\":{\"<certificatename>\":\"$
{TLS_CRT}\"}}"
```

Here,

`<updated-certificate-name>` is the certificate file that contains the updated content.

Example:

Run the following command to update the private key present in the `rsa_private_key_pkcs1.pem` file to the `ocudr-gateway-secret`.

```
TLS_CRT=$(base64 < "rsa_private_key_pkcs1.pem" | tr -d '\n')
kubectl patch secret ocudr-gateway-secret -p "{\"data\":
{\"rsa_private_key_pkcs1.pem\":\"${TLS_CRT}\"}}" -n udr
```

Similarly, you can also update other certificates and keys to the `ocudr-gateway-secret`.

3. Run the following command to remove an existing certificate:

```
kubectl patch secret <secret-name> -p "{\"data\":
{\"<certificatename>\":null}}"
```

Here,

<certificate-name> is the name of the certificate to be removed.

The certificate must be removed when it expires or needs to be revoked.

Example:

Run the following command to remove the CA Root from the ocudr-gateway-secret:

```
kubectl patch secret ocudr-gateway-secret -p "{\"data\":
{\"caroot.cer\":null}}" -n udr
```

Similarly, you can also remove other certificates and keys from the `ocudr-gateway-secret`.

> **Note**
>
> The following are the certificate update and renewal impacts:
>
> - Updating, adding, deleting the certificate, ot terminates all the existing connections gracefully and re-establishes new connections for new requests.
> - When the certificates expires, no new connections are established for new requests, however, the existing connections remain active. After the renewal of the certificates all the existing connections are gracefully terminated. And, new connections are established with the renewed certificates.

B

Rollback Instructions to Revert to 29.519 v15.3.0 for PCF Data

> **Note**
>
> - In a deployed UDR, configure only 1 version of PCF data, either v16.2.0 or v15.3.0. By default, UDR 1.6.0 supports 29.519 v16.2.0 version of PCF data. Having multiple versions create inconsistency in the JSON data stored on UDR.
> - If the user wants UDR to be compatible with PCF data 29.519 v15.3, then follow the instructions shared in this section while deploying UDR. Do not change these instructions if some subscribers are provisioned using any older schema.

> **Important**
>
> A script named as 'rollbackPCFschema_15_3.py' is provided, as part of customer documentation, for this procedure.

To run the script on the CNE bastion:

1. Open the script and edit the following details as per UDR database (udrdb) configurations.

   ```
   mydb=
   ```

2. Run the following command:

   ```
   python rollbackPCFschema_15_3.py
   ```

3. If Python is not installed on the customer setup, then log in to one of the SQL node pods.
4. Run the SQL commands in same order as it is provided in the python script.