

Oracle® Communications

Cloud Native Core, Unified Data Repository

User Guide



Release 25.1.201
G32785-04
October 2025

ORACLE®

Copyright © 2019, 2025, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

Contents

1 Introduction

1.1	References	2
1.2	Architecture	3
1.3	Functionality	5
1.4	Operations Supported	10
1.5	Integration with Other Services	11
1.6	Availability	11

2 Features

2.1	Ingress Gateway Pod Protection Using Rate Limiting	3
2.2	Lifecycle Management (LCM) Based Automation	11
2.3	Traffic Segregation	15
2.4	Support for Post Operation for an Existing Subscription	17
2.5	Support for Kubernetes Resource	19
2.5.1	Support for Dual Stack	20
2.5.2	Network Policies	24
2.5.3	Kubernetes Probes	25
2.6	Suppress Notification	25
2.7	Support for Ingress Gateway and Egress Gateway APIs in CNC Console	29
2.8	Support for Automated Certificate Lifecycle Management	30
2.9	Support for TLS	31
2.10	Support for Provisioning Logs	36
2.11	Error Response and Logging Enhancement	44
2.11.1	Error Response Enhancement	44
2.11.2	Error Logging Enhancement	46
2.12	Ingress Gateway Pod Protection	51
2.13	Support for cnDBTier APIs in CNC Console	58
2.14	Diameter Connection Configuration	59
2.15	Conflict Resolution	61
2.16	Network Function Scoring for a Site	66
2.17	Converged Quota Support	88
2.17.1	Converged Quota Support for UDR	88
2.18	Controlled Shutdown of an Instance	97

2.19	Auto Enrollment and Auto Create Features	103
2.19.1	Auto Enrollment	103
2.19.2	Auto Create	106
2.20	ETag (Entity Tag) support for UDR	108
2.21	Subscriber Export Tool	114
2.21.1	Supports Export of SLF and EIR Subscriber Database	114
2.21.1.1	Exporting SLF Subscriber Database	115
2.21.1.2	Exporting EIR Subscriber Database	115
2.21.2	Support for EXML Format	116
2.21.3	Support for Export of Policy Data in Comma Separated Value (CSV) Format	120
2.21.4	Secure File Transfer Support for Subscriber Export Tool	129
2.21.5	Installing Subscriber Export Tool	131
2.21.6	Customizing Subscriber Export Tool	134
2.21.7	Upgrade and Rollback support for Subscriber Export Tool	153
2.21.8	Monitoring the Subscriber Export Tool Status	154
2.22	Subscriber Activity Logging	156
2.23	Support for 5G EIR Signaling	161
2.23.1	User Equipment Optimization Support for EIR	166
2.23.2	International Mobile Subscriber Identity (IMSI) Fallback Lookup Enhancement for EIR	168
2.23.3	Diameter S13 Interface	170
2.24	Support for LCI and OCI Header	179
2.25	Support for User-Agent Header	183
2.26	Diameter Gateway Pod Congestion Control	187
2.27	Support for Network Exposure Function API	190
2.28	Supports Unstructured Data Storage Function	191
2.29	Support for Default Group ID in SLF	191
2.30	Subscriber Location Function	192
2.30.1	Error Response Enhancement for SLF Lookup Request	193
2.31	Server Header Inclusion in Error Response	195
2.32	Rate Limiting	197
2.33	Overload Handling	198
2.34	Oauth2.0	213
2.35	Multiple Site Redundancy	217
2.36	Migrating Subscriber Data from 4G UDR to 5G UDR	218
2.36.1	Migration Tool	218
2.36.1.1	Installing the Migration Tool	219
2.36.1.2	Configuring Migration Tool	221
2.36.1.3	Using the Migration Tool	233
2.36.2	On-Demand Migration	234
2.36.2.1	Configuring On-Demand Migration	238
2.37	Subscriber Bulk Import Tool	244

2.37.1	Subscriber Bulk Import Tool	245
2.37.1.1	Subscriber Bulk Import Operations	247
2.37.1.2	Bulk Import Operations in iXML Format	248
2.37.1.3	Bulk Import Operations in CSV Format	275
2.37.2	Installing Subscriber Bulk Import Tool	309
2.37.2.1	Configuring Subscriber Bulk Import Tool	314
2.37.2.2	Initiating the Subscriber Bulk Import Tool	328
2.37.2.3	Upgrade and Rollback support for Subscriber Bulk Import Tool	331
2.37.2.4	Sample Subscriber Bulk Import File	332
2.37.3	Monitoring the Subscriber Bulk Import Tool Status	334
2.37.4	Secure File Transfer Support for Subscriber Bulk Import Tool	336
2.38	Alternate Route Service	338
2.39	3gpp-Sbi-Correlation-Info Header	339
2.39.1	3gpp-Sbi-Correlation-Info Header for Ingress Gateway	340
2.39.2	3gpp-Sbi-Correlation-Info Header for Egress Gateway	341
2.40	Automated Test Suite Support	342

3 Configuring Unified Data Repository using CNC Console

3.1	Logging into CNC Console	1
3.2	Egress Gateway	3
3.2.1	Configurable Error Codes	4
3.2.2	Error Code Profile	4
3.2.3	Peer Configuration	6
3.2.4	Peer Set Configuration	7
3.2.5	Routes Configuration	8
3.2.6	SBI Error Action Sets	9
3.2.7	SBI Error Criteria Sets	10
3.2.8	User Agent Header Generation	11
3.3	Ingress Gateway	12
3.3.1	Configurable Error Codes	13
3.3.2	Discard Policy Mapping	13
3.3.3	Error Code Profile	14
3.3.4	Error Code Series	16
3.3.5	OAuth2 Configurations	18
3.3.6	Overload Control Discard Policy	19
3.3.7	Routes Configuration	20
3.3.8	Route Level Rate Limiting	22
3.3.9	Server Header Details	23
3.3.10	User Agent Header Validation	24
3.3.11	Pod Protection	24
3.3.12	Readiness Configuration	28

3.3.13	Controlled Shutdown Error Mapping	28
3.3.14	XFCC Header Validation	29
3.3.15	Congestion Level Configuration	30
3.3.16	Pod Protection By Rate Limiting	31
3.4	cnDBTier	34
3.4.1	cnDBTier Health	34
3.4.2	Backup List	36
3.4.3	cnDBTier Backup Status	37
3.4.4	cnDBTier Version	37
3.4.5	Database Statistics Report	37
3.4.6	Georeplication Recovery	38
3.4.7	GeoReplication Status	39
3.4.8	HeartBeat Status	40
3.4.9	Local Cluster Status	40
3.4.10	On Demand Backup	41
3.5	Access Token Validation	41
3.6	Subscriber Bulk Import Status	43
3.7	Configuration Export & Import	44
3.8	Subscriber Export Tool Status	48
3.9	Global Configurations	48
3.10	Logging Level Configuration	52
3.11	Controlled Shutdown Configurations	56
3.11.1	Operational State	56
3.12	Provisioning	57
3.12.1	Provisioning - NEF Data	58
3.12.2	Provisioning - Profile Data	60
3.12.3	Provisioning - EIR Data	62
3.12.4	Provisioning - PCF Data	65
3.12.5	Provisioning - SLF Data	68
3.12.6	Provisioning - UDM Data	71
3.12.7	Provisioning - Schema Management Data	73
3.13	Service Configurations	75
3.13.1	Subscriber Bulk Import Configurations	76
3.13.2	Data Repository Service	77
3.13.3	Diameter Gateway Service	78
3.13.4	Diameter Gateway Congestion	81
3.13.5	Provisioning Data Repository Service	86
3.13.6	Notify Service	87
3.13.7	On Demand Migration	89
3.13.8	Subscriber Export Tool Configurations	90
3.13.9	Migration Tool Configurations	93
3.14	NF Scoring Configurations	94

3.15	Migration Tool Status	101
3.16	Logout of CNC Console	102

4 Alerts

4.1	Alert Details	4
4.1.1	System Level Alerts	7
4.1.1.1	OcudrSubscriberNotFoundAbove1Percent	7
4.1.1.2	OcudrSubscriberNotFoundAbove10Percent	8
4.1.1.3	OcudrSubscriberNotFoundAbove25Percent	8
4.1.1.4	OcudrSubscriberNotFoundAbove50Percent	9
4.1.1.5	OcudrPodsRestart	9
4.1.1.6	NudrServiceDown	10
4.1.1.7	NudrProvServiceDown	11
4.1.1.8	NudrNotifyServiceServiceDown	12
4.1.1.9	NudrNRFClientServiceDown	13
4.1.1.10	NudrConfigServiceDown	14
4.1.1.11	NudrDiameterProxyServiceDown	15
4.1.1.12	NudrOnDemandMigrationServiceDown	16
4.1.1.13	OcudrIngressGatewayServiceDown	17
4.1.1.14	OcudrEgressGatewayServiceDown	18
4.1.1.15	OcudrDbServiceDown	19
4.1.1.16	OcudrIngressGatewayProvServiceDown	19
4.1.2	Application Level Alerts	20
4.1.2.1	OcudrTrafficRateAboveMajorThreshold	20
4.1.2.2	OcudrTrafficRateAboveMinorThreshold	21
4.1.2.3	OcudrTrafficRateAboveCriticalThreshold	22
4.1.2.4	OcudrTransactionErrorRateAbove0.1Percent	22
4.1.2.5	OcudrTransactionErrorRateAbove1Percent	23
4.1.2.6	OcudrTransactionErrorRateAbove10Percent	23
4.1.2.7	OcudrTrafficRateAboveCriticalThreshold	24
4.1.2.8	OcudrTrafficRateAboveMajorThreshold	25
4.1.2.9	OcudrTrafficRateAboveMinorThreshold	25
4.1.2.10	OcudrTransactionErrorRateAbove0.1Percent	26
4.1.2.11	OcudrTransactionErrorRateAbove1Percent	27
4.1.2.12	OcudrTransactionErrorRateAbove10Percent	27
4.1.2.13	OcudrTransactionErrorRateAbove25Percent	28
4.1.2.14	OcudrTransactionErrorRateAbove50Percent	28
4.1.2.15	OcudrXFCCValidationFailureAbove10Percent	29
4.1.2.16	OcudrXFCCValidationFailureAbove20Percent	29
4.1.2.17	OcudrXFCCValidationFailureAbove50Percent	30
4.1.2.18	DRServiceOverload60Percent	30

4.1.2.19	DRServiceOverload75Percent	31
4.1.2.20	DRServiceOverload80Percent	31
4.1.2.21	DRServiceOverload90Percent	32
4.1.2.22	SLFSucessTxnDefaultGroupldRateAbove1Percent	32
4.1.2.23	SLFSucessTxnDefaultGroupldRateAbove10Percent	32
4.1.2.24	SLFSucessTxnDefaultGroupldRateAbove25Percent	33
4.1.2.25	SLFSucessTxnDefaultGroupldRateAbove50Percent	33
4.1.2.26	OcudrDiameterCongestionCongestedState	34
4.1.2.27	OcudrDiameterCongestionDocState	34
4.1.2.28	DRProvServiceOverload60Percent	35
4.1.2.29	DRProvServiceOverload75Percent	35
4.1.2.30	DRProvServiceOverload80Percent	36
4.1.2.31	DRProvServiceOverload90Percent	36
4.1.2.32	Diameter-Gateway pod congestion Danger of congestion state	37
4.1.2.33	Diameter-Gateway pod CONGESTED state	37
4.1.2.34	OcudrProvisioningTrafficRateAboveMajorThreshold	37
4.1.2.35	OcudrProvisioningTrafficRateAboveCriticalThreshold	38
4.1.2.36	OcudrProvisioningTransactionErrorRateAbove25Percent	39
4.1.2.37	OcudrProvisioningTransactionErrorRateAbove50Percent	40
4.1.2.38	PVCFullForSLFExport	40
4.1.2.39	FailedExtractForSLFExport	40
4.1.2.40	BulkImportTransferInFailed	41
4.1.2.41	ExportToolTransferOutFailed	41
4.1.2.42	BulkImportTransferOutFailed	42
4.1.2.43	PVCFullForXMLBulkImport	42
4.1.2.44	PVCFullForBulkImport	42
4.1.2.45	OperationalStatusCompleteShutdown	43
4.1.2.46	NFScoreCalculationFailed	43
4.1.2.47	PVCFullForUDRExport	44
4.1.2.48	UDRExportFailed	44
4.1.2.49	IngressgatewayPodProtectionDocState	44
4.1.2.50	IngressgatewayPodProtectionCongestedState	45
4.1.2.51	RetryNotificationRecordsMaxLimitExceeded	45
4.1.2.52	UserAgentHeaderNotFoundMorethan10PercentRequest	46
4.1.2.53	EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold	46
4.1.2.54	EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold	47
4.1.2.55	EgressGatewayJVMBufferMemoryUsedAboveCriticalThreshold	47
4.1.2.56	NudrDiameterGatewayDown	48
4.1.2.57	DiameterPeerConnectionsDropped	48
4.1.2.58	IGWSignallingPodProtectionDOCState	49
4.1.2.59	IGWSignallingPodProtectionCongestedState	50

5 KPIs

5.1	Common KPIs for UDR, SLF, and EIR	1
5.1.1	UDR and EIR KPIs	18
5.2	UDR KPIs	23
5.2.1	UDR Notifications	23
5.2.2	On-Demand Migration	26
5.3	SLF KPIs	27
5.3.1	Ingress Gateway Pod Protection Using Rate Limiting	27

6 Metrics

6.1	Common Metrics	1
6.2	UDR Gateways Metrics	2
6.2.1	Ingress Gateway Metrics	2
6.2.2	Egress API Gateway Metrics	11
6.3	nudr-dr-service Metrics	13
6.4	nudr-dr-provservice Metrics	25
6.5	nudr-notify-service Metrics	33
6.6	nudr-diameterproxy-service Metrics	37
6.7	nudr-config-service Metrics	43
6.8	nrf-client Metrics	47
6.9	nudr-migration-service Metrics	52
6.10	nudr-bulk-import-service Metrics	53
6.11	nudr-export-tool service Metrics	57
6.12	nudr-ondemand-migration-service Metrics	59
6.13	app-info Metrics	60
6.14	perf-info Metrics	61
6.15	Diameter Gateway Metrics	62
6.16	Nudr-dbcr-auditor-service	66

A Rollback Instructions for PCF Data to Release v15

B Error Code Catalogue

C Log Attribute Details

D List of Error Codes for Diameter SH Interface

Preface

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Conventions](#)

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

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
CNLB	Cloud Native Load Balancer
CRD	Custom Resource Definition
CNI	Container Network Interface
CSV	Comma Separated Value
cnPCRF	Cloud Native Policy and Charging Rules Function
DAC	Discretionary Access Control
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
GUI	Graphical User Interface
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
LCM	Lifecycle Management
MME	Mobility Management Entity
MSISDN	Mobile Station Integrated Services Digital Network
NADs	Network Attachment Definitions

Table (Cont.) Acronyms and Terminologies

Field	Description
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
NSSF	Oracle Communications Cloud Native Core, Network Slice Selection Function
NSSP	Network Slice Selection Policy
OAM	Operations Administration and Maintenance
OSO	Operational Service Overlay
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 25.1.2xx.

Release 25.1.201 - G32785-04, October 2025

- **General Updates:**
 - Updated the release number to 25.1.201 in the entire document.
 - Updated the procedure for installing the Subscriber Export Tool on multiple sites in the [Installing Subscriber Export Tool](#) section.

Release 25.1.200 - G32785-03, August 2025

- **General Updates:**
 - Updated the threshold values in the [Table 2-2](#).
 - Updated the payload and added a note to update the default values for the fill rate and congestion configurations threshold in the [Ingress Gateway Pod Protection Using Rate Limiting](#) section.
 - Added a note to enable Subscriber Activity Logging with up to 100 keys in the [Subscriber Activity Logging](#) section.

Release 25.1.200 - G32785-02, July 2025

- **General Updates:**
 - Updated the histogram and counter metrics type in the [Metrics](#) section.

Release 25.1.200 - G32785-01, July 2025

- **Feature Updates:**
 - **New Features:**
 - * **Ingress Gateway Pod Protection Using Rate Limiting**
 - * Added the [Ingress Gateway Pod Protection Using Rate Limiting](#) section to describe the feature.
 - * Added the Ingress Gateway Pod Protection Using Rate Limiting feature description in the [Functionality](#) section.
 - * Added the following pages in the CNC Console section:
 - * [Pod Protection By Rate Limiting](#)
 - * [Congestion Level Configuration](#)
 - * Added the following metrics in [Ingress Gateway Metrics](#) section:
 - * oc_ingressgateway_congestion_system_state
 - * oc_ingressgateway_congestion_cpu_state
 - * oc_ingressgateway_http_request_ratelimit_count
 - * oc_ingressgateway_http_request_ratelimit_reject_chain_duration_histogram
 - * oc_ingressgateway_http_request_ratelimit_reject_chain_length_histogram

- * `oc_ingressgateway_http_request_ratelimit_denied_count_total`
 - * `oc_ingressgateway_system_state_duration_percentage`
 - * `oc_ingressgateway_congestion_level_total`
 - * `oc_ingressgateway_congestion_level_bucket_total`
 - * `oc_ingressgateway_congestion_cpu_percentage_bucket`
- * Added the following alerts:
 - * [IGWSignallingPodProtectionDOCState](#)
 - * [IGWSignallingPodProtectionCongestedState](#)
 - * [IGWSignallingPodProtectionByRateLimitRejectedRequest](#)
- * **Lifecycle Management (LCM) Based Automation**
 - * Added the [Lifecycle Management \(LCM\) Based Automation](#) section to describe the feature.
 - * Added the [OSO Alerts Automation](#) section to describe the alert automation process.
 - * Added the Lifecycle Management (LCM) Based Automation feature description in the [Functionality](#) section.
- * **Support for Export of Policy Data in Comma Separated Value (CSV) Format**
 - * Added the [Support for Export of Policy Data in Comma Separated Value \(CSV\) Format](#) section to describe the feature.
 - * Added the Support for Export of Policy Data in Comma Separated Value (CSV) Format feature description in the [Functionality](#) section.
 - * Added `policyDataResources` and `exportFileExpiryTime` parameters in the [Customizing Subscriber Export Tool](#) section.
 - * Added Exported Files Expiry Time and Policy Data Resource for CSV Export fields in the [Subscriber Export Tool Configurations](#) section.
- * **Support for Dual Stack**
 - * Added the [Support for Dual Stack](#) section to describe the feature.
 - * Added the Support for Dual Stack feature description in the [Functionality](#) section.
- **Enhancements:**
 - * **Support for cnDBTier APIs in CNC Console**
 - * Updated the [Support for cnDBTier APIs in CNC Console](#) section to add support for cnDBTier Backup Status in CNC Console.
 - * Added the [cnDBTier Backup Status](#) section in CNC Console.
- **General Updates:**
 - Removed the Memory and Queue as a state parameters in the [Diameter Gateway Pod Congestion Control](#) section.
 - Updated the alert name from `PVCFullForEXMLExport` to [PVCFullForUDRExport](#).
 - Updated the alert name from `EXMLExportFailed` to [UDRExportFailed](#).
 - Added the following metrics in [Ingress Gateway Metrics](#) section:
 - * `oc_oauth_validation_successful_total`

-
- * oc_oauth_validation_failure_total
 - * oc_oauth_certificate_expiryStatus
 - * oc_oauth_certificate_loadStatus
 - * oc_oauth_request_failed_cert_expiry
 - * oc_oauth_keyid_count
- Updated UDR, EIR, and SLF KPIs in the [KPIs](#).
 - The payload for the REST API is updated in the [Support for EXML Format](#) section.
 - Added error codes for Diameter SH Interface in the [List of Error Codes for Diameter SH Interface](#) section.

1

Introduction

Oracle Communications Cloud Native Core Unified Data Repository (UDR) is a key component of the 5G Service Based Architecture. 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:

- HTTP 1.1 and HTTP2 based RESTful interface and APIs to provision other Network Functions
- Provisioning clients to access the 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
- Subscription and Notification feature

Oracle's 5G 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 compliant 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

1.1 References

Refer to the following documents:

- *Oracle Communications Cloud Native Core, Cloud Native Environment User Guide*
- *Oracle Communications Cloud Native Core, cnDBTier User Guide*
- *Oracle Communications Cloud Native Configuration Console User Guide*
- *Oracle Communications Cloud Native Core Automated Test Suite User Guide*
- *Oracle Communications Cloud Native Core, Certificate Management User Guide*
- *Oracle Communications Network Analytics Data Director User Guide*
- *Oracle Communications Cloud Native Core, Network Function Data Collector User Guide*
- *Oracle Communications Cloud Native Core Release Notes*
- *Oracle Communications Cloud Native Core Licensing Information User Guide*
- *Oracle Communications Cloud Native Core Solution Upgrade Guide*
- *Oracle Communications Cloud Native Core Security Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository Network Impact Report*
- *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Provisioning Gateway Installation Guide*
- *Oracle Communications Cloud Native Core, Provisioning Gateway Interface Specification Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository Benchmarking Guide*

1.2 Architecture

The UDR architecture has the following three tiers:

- Connectivity Tier
- Business Tier
- Data Tier

Connectivity Tier

- Ingress API Gateway receives requests from other NFs and forwards them to nudr-drservice of Business Tier.
- Load balances the traffic and provides required authentication using OAuth2.
- Provides TLS support
- Egress API Gateway sends Egress traffic arising from UDR (notifications and NRF management APIs).
- Diameter gateway provides gateway functionality for Ingress and Egress diameter traffic.
- It runs on Kubernetes or OCCNE as a microservice.
- Ingress Gateway microservice: Provides API gateway functionality for ingress traffic received from clients and forwards them to backend microservices. It load balances the requests among all replicas of backend microservices.
- Egress Gateway microservice: Provides API gateway functionality for egress traffic sent to clients and provides configuration to proxy the egress traffic through SCP.

Business Tier

- Provides the business logic of UDR
- It runs on Kubernetes or OCCNE and it has the following microservices:
 - NRF client: handles registration, heartbeat, update, and deregistration with NRF.
 - Nudr-drservice: Handles flexible URI support, run time schema validation, and connects to Data Tier for database operations. It also provides the SLF lookup functionality and N5g-eic service when UDR is deployed in EIR mode as per 3GPP 29.511 specification.
 - Nudr-dr-prov: It is similar to the Nudr-dr microservice but handles only provisioning requests. It also provides provisioning APIs when UDR is deployed in EIR mode for storing EIR equipment status data per PEI or SUPI.
 - Nudr-notify-service: Handles notification messages to PCF and UDM for data subscriptions.
 - Nudr-config service and config-server: Handles all OAM traffic from CNC Console and stores the UDR microservice configurations.
 - Nudr-diameter-proxy: Exposes the diameter endpoint for Sh interface, where diameter clients such as cnPCRF or PCF can connect. It interacts with nudr-notify microservice for notification processing.
 - Nudr-diameter gateway: provides an API gateway functionality for the Diameter messages. It loadbalances the diameter traffic to backend nudr-diameter proxy pods and maintains connections with diameter peer nodes. The diameter notifications i.e Egress messages are also routed through diameter-gateway.

- Nudr-on-demand-migration service: when enabled, it is used to migrate subscribers from 4G UDR to 5G UDR.
- Nudr-perf-info: monitors all microservices in the namespace and collects the metrics such as CPU, memory, load so on.
- Nudr-nrf-client: Is integrated with perf-info to get the load of microservices. It collects the load data from perf-info microservices and sends the information to NRF.
- Nudr-appinfo: monitors all the microservices and databases health and provides an interface to query the health of microservices.
- Nudr-alternatRoute: queries an alternate route for a destination. It enables querying DNS to get an alternate IP for a FQDN and provides a way to configure priority to the alternate routes or virtual IP address.
- Bulk Import Tool: Reads the CSV files and provisions the PCF, SLF, EIR, and 4G policy subscribers from the CSV records. It also provides a way to read iXML files and converts them to CSV format for provisioning. This tool interacts with nudr-dr service for provisioning.
- Migration Tool: Connects to 4G UDR as a diameter client and reads the subscriber data through nudr-diameter gateway. This tool interacts with nudr-dr service for provisioning the subscriber data read from 4G UDR. For more details, see the Migrating Subscriber Data from 4G UDR to 5G UDR section in the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.
- Subscriber Export Tool: The subscriber export tool is part of collection of tools offered by UDR and is used to export the subscriber database from SLF and EIR. The subscriber export tool is deployed in the same namespace of SLF and implemented as a stand alone tool. A new nudr-export-tool microservice is introduced. For more information, see Subscriber Export Tool section in the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

cnData Tier

- Uses Oracle MySQL NDB Cluster, CGE edition as a backend database. This provides High Availability (HA) and georedundancy capabilities.
- Deploys database on either Baremetal, virtualized, Kubernetes platform (kubevirt based), or on CNE as cnDBTier.

Converged Policy DB Solution

UDR has evolved to provide Converged Policy DB Solution to bring its 4G and 5G subscribers together in one common database solution. It provides SOAP based provisioning and Diameter Sh interface as part of the solution.

With this solution:

- Provisioning Gateway is integrated with 5G UDR to support all features required for the converged 4G and 5G DB solution. The Provisioning Gateway needs to be deployed in UDR mode to support this solution:
 - UDR Mode: This mode is applicable when deployed with UDR for converged policy DB solution. In this mode, Provisioning Gateway supports SOAP or XML interface, which is similar to 4G UDR.
- Provisioning Gateway provides provisioning endpoint for SOAP or XML interface with backward compatible interface specifications of 4G UDR.
- Converged data model for 4G and 5G subscriber information is integrated.

- Diameter Sh interface provides access to the subscriber information to 4G PCRF or 5G PCF.
- Migration tool migrates subscribers from 4G UDR to 5G UDR seamlessly by retaining all information.
- On-demand migration feature migrates specific subscribers on-demand when events such as Sh-event, Provisioning event, or NUDR event occurs.

Ingress and Egress Gateway Traffic Management

For more information on Ingress and Egress Gateway Traffic Management, see *Oracle Communications Cloud Native Core, Cloud Native Environment User Guide*.

1.3 Functionality

UDR supports the following functions:

- Storage and retrieval of structured data for exposure
- Integration with cnDBTier
- Integration with Debug Tool
- Alternate route service for NRF in NRF-client
- Configurable API version in the NFProfile
- Flexible URI support: A user can define a new URI for any resource at run time for basic CRUD (create, read, update, and delete) operations on the UDSF (Unstructured Data Storage Function) resources.
- SOAP/XML Interface via Provisioning Gateway. See *Oracle Communications Cloud Native Core, Provisioning Gateway Guide* and *Oracle Communications Cloud Native Core, Provisioning Gateway Interface Specification Guide* for details.
- On-demand migration of 4G subscribers due to Sh event or Provisioning event, or Nudr event
- Bulk Import Tool: To bulk provision PCF, SLF, EIR and 4G Policy subscribers from CSV records. Also supports 4G UDR iXML format for bulk provisioning of 4G Policy Data.
- Migration Tool: To migrate subscribers from 4G UDR
- Provisioning Support using REST/JSON: UDR provides provisioning APIs to create subscribers and to provision different types of NF data (PCF, UDM, SLF, NEF, and EIR). For more details, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.
- Multi-keys: UDR supports multiple keys for a subscriber and provides flexibility to define the new ones
- Runtime Schema Validation: UDR allows users to modify and validate the schema in use for data storage without service restart.
- Schema Versioning: Using this feature, UDR maintains different versions of schema, starting from default version v0 (software version shipped to customers) and as customer upgrades schema, it creates new versions such as v0, v1, v2.
- Schema Management APIs are provided to manage schemas of PCF, UDM, EIR, and SLF data schemas.
- Subscription or Notification: The subscription or notification feature enables consumers to subscribe for and receive notifications when the nudr-drservice receives and processes the subscription request to subscribe the subscriber's resources. Whenever it receives any

update or delete request for subscribers (subscribed for notifications), the nudr-drservice sends an internal signal (HTTP2 POST request) to nudr-notify-service. The nudr-notify-service is responsible to send out the notification requests to the target.

- Supports automatic database and user creation on cnDBTier, as part of Helm installation
- Supports Helm Test to validate the Network Function deployment. For more details, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- Supports Customized Labels, Annotations, and Naming Conventions of Kubernetes Resources in the Helm charts.
- Supports TLS with Ingress and Egress API Gateway
- Supports integration with Provisioning Gateway to receive provisioning updates to SLF
- Supports integration with CNC Console for Manual Subscriber Provisioning on GUI and configuration of all UDR services. Also, SLF groupname configurations can be done on CNC Console GUI
- Supports integration with SCP for the Egress notification traffic
- Supports monitoring of bulk import of subscribers
- Supports Service Mesh Data Plane configurations using Helm for ASM
- Supports logging enhancements for all microservices and traceability using ocLogId
- Supports tracing enhancements to trace requests across all microservices using Jaegar
- Supports IPv6 on the SBI interface (HTTP2 based REST interface).
- Supports multiple sites redundancy with cnDBTier
- Supports server header in the error responses
- Integrated with common NRF client
- Supports OAuth2 authentication with NFInstanceId and Key ID
- Exposes Diameter Sh Interface for 4G Policy data (for all entities), which PCRF uses. It also generates PNR notifications for profile updates
- Uses diameter gateway for Ingress and Egress diameter traffic
- Provides Management Information Base (MIB) for monitoring alerts
- Supports Import and Export of UDR and Provisioning Gateway configuration data through CNC Console.
- Supports overload handling for UDR microservices
- Provides Fault Recovery Procedures for UDR database. See *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- UDSF API: UDR supports basic CRUD operations for UDSF resources as per Rel 15v specification
- Supports integration with CNE services, such as Prometheus or Garafana for metrics, EFK or Kibana for logging, and Jaegar for Tracing
- SLF functionality: Provides the Nudr-groupid-map service towards 5G NRF. Complies with 3GPP release 16, spec 29.504 v16.2.0 and provides provisioning APIs based on SLF Groupname and Nudr-group-id-map

- 5G PCF's AM, SM, UEPolicySet and UsageMonitoring data APIs: UDR supports storage and retrieval of policy data. It is compliant with 3GPP spec v29.519, v16.2 (backward compatible with v15.3.0). See [Rollback to v15.3.0 version of PCF Data](#)
- UDM: UDR supports storage and retrieval of subscription data by UDM. It also supports encryption of stored UDM data. For this, it utilizes the encryption technology of DB Tier (MySQL NDB Cluster).
- UDM API support: Complies with 3GPP spec 29.505 v15.4.0
- Rate limiting for Ingress traffic
- Supports 3GPP SBI correlation header
- Supports configuration of pod disruption budget
- SLF supports provisioning of CHFGrpID, which are not mapped under SLFGrpname
- Supports Default Grp ID for subscribers of SLF
- Supports User Agent Header validation at ingress traffic
- Supports exposure data for NEF
- Supports Diameter Pod Congestion on diameter-gateway
- Supports LCI and OCI Headers in responses and notifications
- Segregates provisioning and signaling traffic using Ingress Gateway instances
- Supports configuration of different TLS certs for provisioning and signaling traffic
- Supports the 5G EIR functionality on the UDR
- Supports 5G EIR bulk provisioning of subscriber data
- Supports Subscriber Activity Logging that provides the network operators to log the subscriber activity for the configured list of subscribers
- Supports exporting of SLF subscriber data into Comma Separated Value (CSV) file format. The Subscriber Export Tool is deployed as a stand-alone tool. The subscriber export tool is by default scheduled for daily runs. The periodicity are daily, weekly, and monthly and these are configurable.
- Supports periodic exporting of EIR database containing all International Mobile Equipment Identity (IMEI) and associated International Mobile Subscriber Identity (IMSI), and Mobile Station Integrated Services Digital Network (MSISDN) using the subscriber export tool. It provides automated and on-demand export of EIR database.
- The subscriber export tool provides the capability to securely transfer the exported data to a remote server using Secure File Transfer Protocol (SFTP). It also supports monitoring of the status of exported data transfer through CNC Console.
- The remote file transfer for subscriber bulk import for EIR provides the capability for securely transferring the files from the remote server using SFTP. This feature supports transferring-in the Provisioning Database Interface (PDBI) import files and transferring-out the result log files to a remote server.
- ETag (Entity Tag): When two different Cloud Native Policy and Charging Rules Function (cnPCRF) is connected to UDR in a georedundant setup and both the PCRF are updating the UsageMonData for the same subscriber simultaneously, there is possibility of data loss as the update can overwrite each other. ETag support for UDR helps to make sure that the update is successful at UDR only when the consumer has the latest set of data, which is determined by the ETag value received at UDR.

- Etag Notifications: UDR sends updated ETag notifications header to the consumer about the current state of the usage monitoring data. This enables the consumer to update the ETag values in the subsequent update operations.
- Controlled Shutdown on UDR: UDR supports controlled shutdown feature to provide the partial or complete isolation of the site from the network so that the operator can perform necessary recovery procedures when required.
- Auto Enrollment: Consumer NFs such as cnPCRF need UDR to enroll the subscriber record with policy data sent on the N36 interface. This is required for subscribers in roaming scenarios, where cnPCRF does a PATCH operation to update the subscriber data. Auto enrollment enables UDR to auto enroll the subscriber if the subscriber record does not exist. Auto enrollment also supports UDR in creating mandatory attribute smPolicySnssaiData with configured default values when NF such as cnPCRF does not include smPolicySnssaiData attribute in sm-data.
- Aspen Service Mesh (ASM) supports the following enhancements for UDR:
 - Transmission Control Protocol (TCP) Connection in destination rules
 - TCP connection and The Hypertext Transfer Protocol (HTTP) stream in envoy filters
 - HTTP route in virtual service
- The Network Function (NF) Scoring feature calculates the score for a site based on NF specific factors such as alerts. The NF Scoring feature helps the operator to determine the health of a site as compared to other sites. Comparing the NF scores within or across the sites helps the customers to select the appropriate site.
- Converged Quota Support for Provisioning Interface feature updates the data model to support storage of 4G policy quota and dynamic quota under sm-data/umData and sm-data/umDataLimits respectively from VSA. This feature enables cnPCRF to access and manage 4G and 5G quota from umData and umDataLimits. The following interfaces are compatible with the converged data model:
 - Provisioning interface using SOAP/XML
 - Bulk provisioning using file import in iXML and CSV format
 - N36 signaling interface
 - Provisioning Interface using REST/JSON
 - On-Demand Migration
 - Diameter-Sh
- Network Policies are an application-centric construct that allows you to specify how a pod is allowed to communicate with various network entities. Network Policies creates pod-level rules to control communication between the cluster's pods and services. It determines which pods and services can access one another inside the cluster.
- The Diameter Connection Configuration triggers a Connection Establishment Request (CER) from the Diameter Gateway pod to each of the configured diameter endpoints. This feature enables the distribution of the diameter connection in a controlled manner by balancing the diameter traffic evenly between the Diameter Gateway pods. This is achieved by adding peer node details as a configuration in the Diameter Gateway.
- Conflict Resolution: In a multisite deployment, UDR updates subscriber data on all the sites during replication down time. When the replication is restored and if there are any conflicts in the subscriber data updates, they are automatically resolved in all the database tables by MySQL NDB Cluster in cnDBTier. However, in some cases, automatic resolution by MySQL causes subscriber data conflicts and anomalies in the database model. This feature provides an application-defined resolution to resolve the subscriber data conflicts by adding a timestamp column to determine whether an update is required on the replica.

- **Error Response Enhancement for SLF Lookup Request:** SLF supports the “cause” attribute in the problem details for all error responses as per 3GPP 29.504 specification. The “cause” attribute in the problem details object of the HTTP error response payload is used to indicate the application-related information for that specific occurrence of the error. This enables the HTTP client to take appropriate action. The error response is enhanced to support error causes scenario for SLF lookup requests.
- **Migration Tool Support for Converged Quota:** The migration tool is enhanced to support converged quota and it also supports monitoring and pausing the migration tool using REST API and CNC Console.
- **Support for EXML Format:** This feature allows to export the subscriber data in EXML format, which is compatible with 4G OCUDR export format. It supports the export of 4G policy data (VSA and umData/umDataLimits) in EXML format from 5G UDR to 4G OCUDR using the subscriber export tool.
- **User Equipment Optimization support for Equipment Identity Register (EIR):** As per 3GPP standards, all User Equipment (UE) in the operators network is declared within the EIR database as Whitelisted, Greylisted, or Blacklisted. The UEs that are not declared in the EIR database are categorized Blacklisted by default. As all UE Permanent Equipment Identifier (PEI) and International Mobile Equipment Identity (IMEI) cannot be provisioned in EIR database, this feature allows the operators to configure a default response for unknown UE.
- **The provisioning logs enhances the provisioning logging functionality** by enabling a logging type that is independent of the logging level. This feature helps in collecting the subscriber provisioning logs by enabling the provisioning logging mechanism. It also enables the operators to log the 5G subscriber provisioning requests for create, update, and delete operations.
- **Error Response Enhancement:** The error responses used to contain only the error description in the details field, which was insufficient to troubleshoot any error. Using the enhanced error response mechanism, UDR sends additional pieces of information such as server FQDN, NF service name, vendor name, and error ID in the details field of the payload to help in the quick identification of the source of an error response.
- **Ingress Gateway Pod Protection:** This feature protects the Ingress Gateway pods from overloading due to uneven traffic distribution, traffic bursts, or congestion. During overload conditions, the Ingress Gateway pods may undergo stability issues. As a front end microservice for HTTP traffic, it is important for Ingress Gateway to have pod protection implemented.
- **UDR supports automation of certificate lifecycle management in integration with Oracle Communications Certificate Manager (OCCM).** This enables to automatically create, renew, and delete certificates for a given CA, with the possibility to track previously created certificates and renew/delete them when required.
- **Suppress Notification feature** enables cnUDR to store the User-Agent header received in the POST request from cnPCRF in the subscription table. cnUDR compares the User-Agent header received during an update operation from cnPCRF with the stored User-Agent header, if the User-Agent header matches, then the notification is suppressed. The notification is sent if the User-Agent header does not match or if there is no User-Agent header in the update request.
- **The configuration for common service APIs** was earlier supported only using REST API. With the implementation of Common Service APIs in CNC Console, UDR now supports the configuration of Ingress Gateway and Egress Gateway parameters using the CNC Console.

- UDR can enhance security by supporting TLS 1.3 for all consumer NFs, producer NFs, the Data Director, SBI Interfaces, and any interfaces that previously supported TLS 1.2. UDR uses HTTPS with TLS encryption to establish secure connections with these components.
- Error Logging Enhancement: With this feature, UDR sends additional information to the existing "ERROR" log messages to identify the cause of the issue and minimize the troubleshooting time. Additional attributes are added to the existing ERROR logs which gets populated with appropriate values during failure scenarios.
- International Mobile Subscriber Identity (IMSI) Fallback Lookup Enhancement feature enables EIR to return equipment status as WHITELISTED, BLACKLISTED, or GREYLISTED for the matched IMEI in the EIR database.
- Diameter S13 Interface feature provides support between EIR and Mobility Management Entity (MME) to retrieve the UE equipment status of the subscriber.
- Traffic Segregation feature provides an option for traffic segregation at UDR based on traffic types. Within a Kubernetes cluster, traffic segregation can divide applications or workloads into distinct sections such as Operations Administration and Maintenance (OAM), Service Based Interface (SBI), Kubernetes control traffic, and so on. The new functionality ensures that critical networks are not cross-connected or sharing the same routes, thereby preventing network congestion.
- Post Operation for an Existing Subscription feature enables UDR to support POST request that overwrites the existing subscription.
- The Lifecycle Management (LCM) Based Automation feature optimizes the deployment or upgrade steps. This is achieved by automating service account creation that enables you to create user-defined service account automatically without performing any manual steps.
- Using the dual stack mechanism, applications or NFs can establish connections with pods and services in a Kubernetes cluster using IPv4 or IPv6.
- Support for Policy Data Export in Comma Separated Value (CSV) Format enables Subscriber Export Tool to export the 5G and 4G subscriber policy data which includes the profile data and policy data (am-data, sm-data, and ue-policy-set) in CSV file format from cnUDR. The converted subscriber data in CSV file format is used by the Subscriber Bulk Import Tool to import the data along with its policy data on an another instance of cnUDR.
- With the implementation of Ingress Gateway Pod Protection Using Rate Limiting feature, rate-limiting mechanism is applied for Ingress Gateway pods. This mechanism allows pods to process a predefined number of requests. When the request rate exceeds the threshold, the pods take action to protect themselves. They either reject additional requests with a custom error code or allow them, depending on the configuration.

1.4 Operations Supported

UDR supports the following operations:

Subscriber Profile Related Operations for Provisioning

- **Create Subscriber:** Provisions a subscriber with the given set of keys
- **Update Subscriber:** Updates the subscriber information
- **Get subscriber:** Retrieves the subscriber information completely
- **Delete Subscriber:** Deletes the subscriber information and related data

Network Function Data Related Operations as per 3GPP

- **Insert Data:** Creates a subscriber if not present and inserts the specific data (policy_data/udsf_data/udm_data/SLF Data/exposure_data).
- **Update Data:** Creates a subscriber if not present and updates the complete data for a particular subscriber as given in request.
- **Patch Data:** Supports patch update of a specific data or parts of it. If not present, it adds the data. This operation is valid only when subscriber is already present.
- **Get Data:** Retrieves the requested data for a particular subscriber.
- **Delete Data:** Deletes the requested data for a particular subscriber.
- **Subscription Request:** Subscribes to a subscriber resources and get notified in case of updates.
- **Notification:** UDR generates notifications when there are updates to subscriber resources subscribed in previous request. These notifications are sent to notification URI received in subscription request.

1.5 Integration with Other Services

UDR is integrated with:

- CNE Services: Such as Prometheus or Grafana for metrics, UDR is integrated with EFK or Kibana for logging and Jaeger for tracing.
- Provisioning Gateway and 4G Policy Data: For receiving provisioning updates to SLF. For more information, see *Oracle Communications Cloud Native Core, Provisioning Gateway Guide*.
- SCP for Egress Notification Traffic. For more information, see the **Configuring User Parameters** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- Cloud Native Configuration Console (CNCC): It provides a GUI interface for subscriber provisioning and configuration of UDR services. The CNCC GUI allows UDR administrators to configure SLF groupname.
- Debug Tool: For more information, see the Troubleshooting chapter of *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
- cnDBTier: For more information, see the *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

1.6 Availability

Cloud Native Core Unified Data Repository availability is dependent on many factors. This network function is designed to achieve 99.999% availability, pursuant to the applicable Telecommunications Industry Association TL9000 standards, with the following deployment requirements:

- Deploy on a Cloud Native Environment with at least 99.999% Availability
- Deploy with $n + k$ application redundancy, where k is greater than or equal to 1
- Maintain production software within $n-3$ software releases where n = the current GA release
- Apply bug fixes, critical patches, and configuration recommendations provided by Oracle in a timely manner
- Maintain fault recovery procedures external to the applications for reconstruction of lost or altered files, data, programs, or cloud native environment

- Install, configure, operate and maintain in accordance with Oracle's applicable installation, operation, administration, and maintenance specifications
- Maintain active support contract and provide access to the deployed UDR and your personnel to assist Oracle in addressing any outage

UDR availability is measured over the period of each calendar year and is calculated as follows:

Table 1-1 Measuring UDR Availability

Item	Description
Planned Product Availability	Product available time in each month less Excluded Time (defined below) in each month
Actual Product Availability	Planned Product Availability less any Unscheduled Outage
Product Availability Level	$(\text{Actual Product Availability across all Production instances} \div \text{Planned Product Availability across all Production instances}) \times 100$

Note

Excluded Time means:

- Scheduled maintenance time
- Lack of power or backhaul connectivity, except to the extent that such lack of backhaul connectivity was caused directly by the CNC NF
- Hardware failure
- Issues arising out of configuration errors or omissions
- Failures caused by third party equipment or software not provided by Oracle
- Occurrence of any event under Force Majeure
- Any time associated with failure to maintain the recommended architecture and redundancy model requirements above

2

Features

This section explains Oracle Communications Cloud Native Core, Unified Data Repository (UDR) features.

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.

The following table lists the features applicability for UDR, SLF, and EIR:

Table 2-1 Features Applicability

Features	UDR	SLF	EIR
#unique_51	Yes	Yes	Yes
Ingress Gateway Pod Protection Using Rate Limiting	No	Yes	No
Lifecycle Management (LCM) Based Automation	Yes	Yes	Yes
Support for Dual Stack	Yes	Yes	Yes
Traffic Segregation	Yes	Yes	Yes
Support for Post Operation for an Existing Subscription	Yes	NA	NA
Suppress Notification	Yes	NA	NA
Support for Ingress Gateway and Egress Gateway APIs in CNC Console	Yes	Yes	Yes
Support for Automated Certificate Lifecycle Management	Yes	Yes	Yes
Support for TLS	Yes	Yes	Yes
Support for Provisioning Logs	Yes	No	No
Error Response Enhancement	Yes	Yes	Yes
Error Logging Enhancement	Yes	Yes	Yes
Ingress Gateway Pod Protection	Yes	Yes	Yes
Support for cnDBTier APIs in CNC Console	Yes	Yes	Yes
Diameter Connection Configuration	Yes	NA	Yes
Network Policies	Yes	Yes	Yes
Conflict Resolution	Yes	Yes	Yes
Network Function Scoring for a Site	No	Yes	No
Converged Quota Support	Yes	NA	NA
Kubernetes Probes	Yes	Yes	Yes
Controlled Shutdown of an Instance	No	Yes	No

Table 2-1 (Cont.) Features Applicability

Features	UDR	SLF	EIR
Auto Enrollment	Yes	NA	NA
Auto Create	Yes	NA	NA
ETag (Entity Tag) support for UDR	Yes	NA	NA
Subscriber Export Tool	Yes	Yes	Yes
Supports Export of SLF and EIR Subscriber Database	NA	Yes	Yes
Exporting SLF Subscriber Database	NA	Yes	NA
Exporting EIR Subscriber Database	NA	NA	Yes
Support for EXML Format	Yes	NA	NA
Support for Export of Policy Data in Comma Separated Value (CSV) Format	Yes	NA	NA
Secure File Transfer Support for Subscriber Export Tool	Yes	Yes	Yes
Subscriber Activity Logging	Yes	Yes	Yes
Support for 5G EIR Signaling User Equipment Optimization Support for EIR International Mobile Subscriber Identity (IMSI) Fallback Lookup Enhancement for EIR Diameter S13 Interface	NA	NA	Yes
Support for LCI and OCI Header	Yes	Yes	Yes
Support for User-Agent Header	Yes	Yes	Yes
Diameter Gateway Pod Congestion Control	Yes	NA	Yes
Support for Network Exposure Function API	Yes	NA	NA
Support for Default Group ID in SLF	NA	Yes	NA
Subscriber Location Function	NA	Yes	NA
Rate Limiting	Yes	Yes	Yes
Overload Handling	Yes	Yes	Yes
OAuth2.0	Yes	Yes	Yes
Multiple Site Redundancy	Yes	Yes	Yes
Migrating Subscriber Data from 4G UDR to 5G UDR Migration Tool On-Demand Migration	Yes	NA	NA
Subscriber Bulk Import Tool	Yes	Yes	Yes
Bulk Import Operations in iXML Format	Yes	NA	NA
Bulk Import Operations in CSV Format	NA	Yes	Yes
Secure File Transfer Support for Subscriber Bulk Import Tool	Yes	Yes	Yes
Alternate Route Service	Yes	Yes	Yes
Automated Test Suite Support	Yes	Yes	Yes

2.1 Ingress Gateway Pod Protection Using Rate Limiting

The Ingress Gateway microservice manages all incoming traffic from consumer Network Functions (NFs). However, it is vulnerable to performance issues during an unexpected traffic spikes, uneven distribution, or bursts caused by network fluctuations. To ensure the system stability and prevent potential network outages, a protection mechanism for the Ingress Gateway is necessary.

With the implementation of this feature, rate limiting mechanism is applied for Ingress Gateway pods. This mechanism allows pods to process a predefined number of requests. When the request rate exceeds the threshold, the pods take action to protect themselves. Depending on the configuration, the pods either reject the additional requests with a custom error code or allows the request.

Each Ingress Gateway pod has its own Pod Protection Policer, which is applied individually. When UDR rejects the requests, it is expected that the consumer NFs reroute to alternate UDRs.

Note

Ingress Gateway Pod Protection Using Rate Limiting feature supports only SLF mode. For UDR and EIR support refer to [Ingress Gateway Pod Protection](#).

The Ingress Gateway's configuration for this feature includes:

- **Fill Rate:** This sets the maximum number of requests a pod can handle in a defined time interval. It can be customized for multiple routes by specifying the route path and percentage of requests.
- **Congestion Configuration:** This allows the configuration of congestion levels based on CPU resource utilization.
- **Denied Request Actions:** This defines the action to be taken when the fill rate is exceeded. The options are 'CONTINUE', allowing requests to be processed, or 'REJECT', which rejects requests with a specified error code.

The fill rate represents the system's capacity when traffic is evenly distributed over one second. However, in bursty traffic scenarios, the actual failure rate might be higher, even if the average traffic stays within the limit. Ingress Gateway also rejects excess requests in the following situations:

- When the congestion state configuration is not found, it rejects the request with a default error.
- If the denied request actions configuration is blank, it rejects and uses a default error for the rejection.
- If the congestion level does not match with any configuration, the default action is to reject the request.

Congestion level is detected every interval (`congestionconfig.refreshInterval`) and the action is decided based on the level in that interval. It is possible that the CPU levels might vary between these intervals causing the congestion levels to switch frequently. To understand the detected congestion level, see the metric `oc_ingressgateway_congestion_level_bucket_total`.

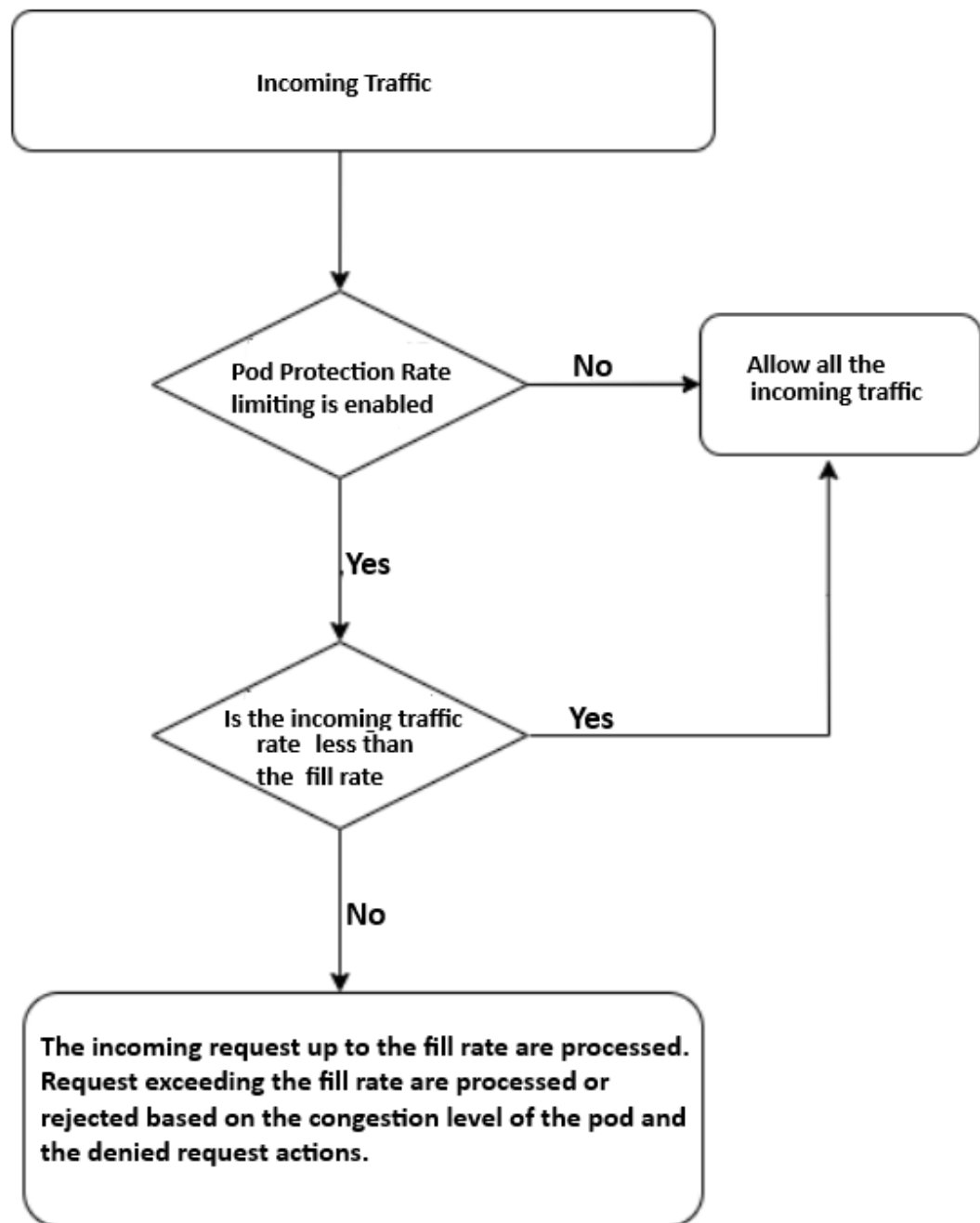
The following table describes the configuration for different congestion levels of the pods:

Table 2-2 Congestion Levels of Pods

Level Value	Threshold and Level	Description
1	Abatement Threshold: 65 Onset Threshold: 70 Level Name: Normal	The default congestion level of pod is "0" (0-64), when the CPU is less than the abatement of Normal (65). The pod remains in Normal state until: • CPU is less than 80 • CPU is more than 70
2	Abatement Threshold: 75 Onset Threshold: 80 Level Name: Danger of congestion	Pod remains in Danger of Congestion state until: • CPU is less than 90 • CPU is more than 80
3	Abatement Threshold: 85 Onset Threshold: 90 Level Name: Congested	The pod moves to Congested state when the CPU level is more than 90.

The following diagram explains the Ingress Gateway Pod Protection Using Rate Limiting mechanism:

Figure 2-1 Ingress Gateway Pod Protection Using Rate Limiting



1. Consumer NFs send traffic to the Ingress Gateway microservice.
2. Ingress Gateway pod checks if the Pod Protection Using Rate Limiting feature is enabled or not.
 - a. If the feature is enabled, Ingress Gateway compares the incoming traffic rate with the fill rate that is defined for the Rate Limiting feature.
 - i. If the incoming request rate is below the fill rate, then all the incoming traffic are processed.

- ii. If the incoming request rate exceeds the fill rate, then requests up to the fill rate are processed. The requests beyond the fill rate are processed or rejected based on the congestion level of the pod and the configuration of the denied request actions.

Consider the following examples:

- If the fill rate is configured as 1000, then:
 - if the incoming Transaction Per Second (TPS) is 1100, congestion level is Normal, and action is configured as CONTINUE, all the 1100 requests are processed.
 - if the incoming TPS is 1300, congestion level is Danger of Congestion, and action is configured as CONTINUE, all the 1300 requests are processed.
 - if the incoming TPS is 1500, congestion level is Congested, and action is configured as REJECT, 1000 requests are processed successfully and the 500 requests are rejected with the configured error code.

Managing Ingress Gateway Pod Protection Using Rate Limiting

Prerequisites

Note

- It is recommended to enable only one of the Ingress Gateway pod protection feature. You can disable the Ingress Gateway Pod Protection before proceeding to this feature. For more information about disabling the existing pod protection feature, see [Ingress Gateway Pod Protection](#). The feature Ingress Gateway Pod Protection will be deprecated in the upcoming releases.
- It is recommended to use REST API to enable the Ingress Gateway Pod Protection Using Rate Limiting feature.

Enable

You can enable the Ingress Gateway Pod Protection Using Rate Limiting feature using Helm Configuration or CNC Console:

- **Enable using Helm Configuration:** You must modify the `ocudr-custom-values.yaml` file as follows before installing UDR:
 - Change the mode of the `podProtectionByRLConfigMode` parameter to HELM.
 - Enable the `podProtectionByRateLimiting.enabled` parameter to true.
 - Change the mode of the `congestionConfigMode` parameter to HELM.

```
podProtectionByRLConfigMode: REST
podProtectionByRateLimiting:
  enabled: false
  fillRate: 2800           # Indicates tokens to be added in 1000ms
  bucketAllocation: 25    # Bucket allocation is in percentage
  routes: # To keep routes empty, configure it as mentioned -> routes: []
    - id: 1
      methods:
        - GET
      path: /nudr-group-id-map/**
      percentage: 100      # FillRate for this route in % of the
```

```

aggregateFillRate, 40% would be 400 tokens per 1000ms
# Actions to be taken based on the priority for all the requests denied
by the rate limiting
  errorCodeProfile: error429
# Actions to be performed for denied requests based on SBI Message
Priority and Pod Local Congestion Level
  deniedRequestActions:
    - id: 1
      congestionLevel: 0 # Reference to
congestionConfig.levels[].value
      action: CONTINUE # Possible values -
REJECT, CONTINUE
    - id: 2
      congestionLevel: 1 # Reference to
congestionConfig.levels[].value
      action: CONTINUE # Possible values -
REJECT, CONTINUE
    - id: 3
      congestionLevel: 2 # Reference to
congestionConfig.levels[].value
      action: REJECT # Possible values -
REJECT, CONTINUE
      errorCodeProfile: error429 # Reference to
errorCodeProfiles[].name
    - id: 4
      congestionLevel: 3 # Reference to
congestionConfig.levels[].value
      action: REJECT # Possible values -
REJECT, CONTINUE
      errorCodeProfile: error503 # Reference to
errorCodeProfiles[].name
      priorityHeaderName: 3gpp-Sbi-Message-Priority # name of the header to
use for priority.
      defaultPriority: 24 # Default message
priority to be used in absence of priority header.

congestionConfigMode: REST # Congestion
Configuration Mode. Possible value - HELM, REST
congestionConfig: # congestion level calculation configuration
  refreshInterval: 500
  levels: # levels defines different congestion levels. Max: 10 levels are
supported.
    - value: 1 # level numeric value.
Possible values 1-10. 0 is reserved for default level.
      name: Normal # pretty name
      resources: # resource thresholds.
        - name: CPU # Possible values - CPU
          onset: 70 # onset threshold
          abatement: 65 # abatement threshold
    - value: 2 # level numeric value.
Possible values 1-10. 0 is reserved for default level.
      name: Danger Of Congestion # pretty name
      resources: # resource thresholds.
        - name: CPU # Possible values - CPU
          onset: 80 # onset threshold
          abatement: 75 # abatement threshold

```

```

- value: 3                                # level numeric value.
Possible values 1-10. 0 is reserved for default level.
name: Congestion                          # pretty name
resources: # resource thresholds.
- name: CPU                               # Possible values - CPU
onset: 90                                 # onset threshold
abatement: 85                             # abatement threshold

```

For more information about the parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

- **Enable using CNC Console:** You can enable the feature by setting the Enabled flag to true and configure the other parameters as mentioned in the [Pod Protection By Rate Limiting](#) in the CNC Console.

Configure

- **Configure using REST API:** You can configure this feature using REST API as follows:
 - Set the enabled parameter to true and perform the PUT operation on the `http://<ocudr-nudr-config:5001>/udr/nf-common-component/v1/igw-sig/podProtectionByRateLimiting/node`

```

{
  "routes": [
    {
      "id": 1,
      "path": "/nudr-group-id-map/**",
      "methods": [
        "GET"
      ],
      "percentage": 100
    }
  ],
  "enabled": true,
  "fillRate": 2800,
  "defaultPriority": 24,
  "errorCodeProfile": "error429",
  "priorityHeaderName": "3gpp-Sbi-Message-Priority",
  "deniedRequestActions": [
    {
      "id": 1,
      "action": "CONTINUE",
      "congestionLevel": 0
    },
    {
      "id": 2,
      "action": "CONTINUE",
      "congestionLevel": 1
    },
    {
      "id": 3,
      "action": "REJECT",
      "congestionLevel": 2,
      "errorCodeProfile": "error429"
    }
  ]
}

```

```

        "id": 4,
        "action": "REJECT",
        "congestionLevel": 3,
        "errorCodeProfile": "error503"
      }
    ]
  }

```

- By default the supported error codes are 429 and 503. To configure more error codes, you must configure using the `errorcodeprofiles` API on the `http://<ocudr-nudr-config:5001>/udr/nf-common-component/v1/igw-sig/errorcodeprofiles`

```

[
  {
    "name": "error429",
    "errorCode": 429,
    "errorCause": "Too many reequets",
    "errorTitle": "Too many reequets",
    "redirectURL": "",
    "retry-after": "",
    "errorDescription": "Too many reequets"
  },
  {
    "name": "error503",
    "errorCode": 503,
    "errorCause": "Backend not able to handle traffic",
    "errorTitle": "Backend not able to handle traffic",
    "redirectURL": "",
    "retry-after": "",
    "errorDescription": "Backend not able to handle traffic"
  }
]

```

- You can change the threshold levels for the CPUs by using the `congestionConfig` API on the `http://<ocudr-nudr-config:5001>/udr/nf-common-component/v1/igw-sig/congestionConfig/node` as follows:

```

{
  "levels": [
    {
      "name": "Normal",
      "value": 1,
      "resources": [
        {
          "name": "CPU",
          "onset": 70,
          "abatement": 65
        }
      ]
    },
    {
      "name": "Danger Of Congestion",
      "value": 2,
      "resources": [
        {

```

```

        "name": "CPU",
        "onset": 80,
        "abatement": 75
      }
    ],
  },
  {
    "name": "Congestion",
    "value": 3,
    "resources": [
      {
        "name": "CPU",
        "onset": 90,
        "abatement": 85
      }
    ]
  }
],
"refreshInterval": 500
}

```

For more information about this API, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

- **Configure using CNC Console:**
 - Configure the Error code Profiles, as described in [Error Code Profile](#).
 - Create or update the Congestion Level Configuration, as in [Congestion Level Configuration](#).

After installing or upgrading SLF, you must update the default values for the fill rate and congestion configurations threshold using the CNC Console or REST API as follows:

- Update the fill rate to 2800
- Update the configurations threshold values as follows:

Table 2-3 Threshold Level

Level Name	Threshold
Normal	Onset: 70 Abatement: 65
DOC	Onset: 80 Abatement: 75
Congested	Onset: 90 Abatement: 85

Observability

Metrics

The following metrics are added for this feature:

- `oc_ingressgateway_congestion_system_state`
- `oc_ingressgateway_congestion_cpu_state`

- `oc_ingressgateway_http_request_ratelimit_count`
- `oc_ingressgateway_http_request_ratelimit_reject_chain_duration_histogram`
- `oc_ingressgateway_http_request_ratelimit_reject_chain_length_histogram`
- `oc_ingressgateway_http_request_ratelimit_denied_count_total`
- `oc_ingressgateway_system_state_duration_percentage`
- `oc_ingressgateway_congestion_level_total`
- `oc_ingressgateway_congestion_level_bucket_total`
- `oc_ingressgateway_congestion_cpu_percentage_bucket`

For more information about the metrics, see [Ingress Gateway Metrics](#).

Alerts

The following alerts are added for this feature.

- [IGWSignallingPodProtectionDOCState](#)
- [IGWSignallingPodProtectionCongestedState](#)
- [IGWSignallingPodProtectionByRateLimitRejectedRequest](#)

KPIs

The following KPIs are added for this feature:

- Congestion Level
- Allowed/Rejected Requests
- Allow/Reject Requests per Pod
- Overall CPU Usage

For more information about KPIs, see [Ingress Gateway Pod Protection Using Rate Limiting](#).

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.2 Lifecycle Management (LCM) Based Automation

This feature optimizes the deployment or upgrade steps. This is achieved by automating service account creation that enables you to create user-defined service account automatically without performing any manual steps. The automation includes:

- Creating Service Account: An automated resource service account creation has been introduced in this release to streamline the management of Kubernetes resources through Helm charts.
- Applying NF Alert Configuration rules: UDR leverages the `oso-alr-config` Helm chart, introduced as part of the `OSO` package, to apply alert rules using Helm upgrades. While

the oso-alr-config Helm chart is deployed automatically during the OSO installation, the UDR performs Helm upgrades to apply or update the required alert rules. Both manual and automated configurations are supported.

Operational Service Overlay (OSO) Alerts Automation

Alerts are automated by using Helm upgrade command with the Helm chart provided as part of OSO software package. For more information, see [OSO Alerts Automation](#).

Managing LCM Based Automation

Enable

You can enable this feature using the Helm.

Automating Service Account, Role and Role Binding

To automate the creation of service account, role, and role binding, enable the following parameters:

- Global Flag (`autoCreateResources.enabled`): Controls the overall automation of resource creation. This flag is disabled by default and must be set to true to enable automation.
- Resource-specific Flag (`serviceAccounts.create`): Controls ServiceAccount creation at the resource level. This flag is enabled by default and make sure it is set to true alongside the global flag to enable ServiceAccount automation. This flag is conditional on the global flag and will only take effect if the global flag is set to true. If the flag is disabled, you must create the service accounts manually. The role and role binding resources are created along with the service account as part of this automation.

```
global:
# Flag for Auto-creation of resources and is disabled by default.
# - For now the resources include Service Account configuration
#
autoCreateResources:
  enabled: true

  serviceAccounts:
    create: true # This
internal flag controls whether Service Accounts should be created
automatically.
    accounts:
      - serviceAccountName: *serviceAccountName # Name of
UDR ServiceAccount.
        type: UDR # Type
determines RBAC rules to be constructed in the template
```

The service account automation is disabled by default in the custom values yaml file. Perform the following steps to enable the service account automation:

Note

You must also perform the following steps during the upgrade procedure.

1. Provide the `serviceAccountName` in the custom values yaml file to create the service account.

2. Enable the `autoCreateResources.enabled` and `serviceAccounts.create` parameters in the global section of the custom values yaml file.
3. Perform the Helm installation. The service account will be created with the name provided in the custom values yaml file.

The existing `serviceAccountName` field in the custom values yaml file is used for service account automation to minimize the changes in the custom values yaml file.

```
#Provide service account name, as part of automation given with given service
account name service account will be created
serviceAccountName: &serviceAccountName udr-sa
```

```
autoCreateResources:
  enabled: true

  serviceAccounts:
    create: true # This internal
flag controls whether ServiceAccounts should be created automatically.
    accounts:
      - serviceAccountName: *serviceAccountName # Name of UDR
ServiceAccount.
        type: UDR # Type
determines RBAC rules to be constructed in the template
```

The role and role binding resources are created with the release name without overwriting the manual resources as part of this automation. The service account automation is supported only for Helm. The following table provides the service account creation behavior using different combinations.

Table 2-4 Service Account Creation using Different Combinations

Parameter	serviceAccountName	Result
<code>autoCreateResources.enabled: true</code> <code>serviceAccounts.create: true</code>	Provided	Service account is created or updated with the provided <code>serviceAccountName</code> .
<code>autoCreateResources.enabled: true</code> <code>serviceAccounts.create: true</code>	Not provided	Service account is created or updated with <code>.Release.name</code> .
<code>autoCreateResources.enabled: true</code> <code>serviceAccounts.create: false</code>	Provided	Service account is not created.
<code>autoCreateResources.enabled: true</code> <code>serviceAccounts.create: false</code>	Not provided	The deployment fails. The <code>serviceAccountName</code> is mandatory.
<code>autoCreateResources.enabled: false</code> <code>serviceAccounts.create: true</code> or <code>false</code>	Provided	The service account is not created. It must be created manually.

Table 2-4 (Cont.) Service Account Creation using Different Combinations

Parameter	serviceAccountName	Result
autoCreateResources.enabled: false serviceAccounts.create: true or false	Not Provided	Service account is created or updated with .Release.name.

The Subscriber Bulk Import Tool, Subscriber Export Tool, and Migration Tool charts are modified to support the automatic creation of service account and role and role binding.

Configure

Automating Service Account, Role and Role Binding

You can configure the service accounts automation using Helm as follows:

```
#Provide service account name, as part of automation given with given service
account name service account will be created
serviceAccountName: &serviceAccountName udr-sa
```

```
autoCreateResources:
  enabled: true
```

```
  serviceAccounts:
    create: true # This internal
flag controls whether ServiceAccounts should be created automatically.
    accounts:
      - serviceAccountName: *serviceAccountName # Name of UDR
ServiceAccount.
        type: UDR # Type
determines RBAC rules to be constructed in the template
```

Table 2-5 Parameter Details

Parameter Details	Description
Parameter name: serviceAccountName Default Value: NA	The service account is created using this name.
Parameter name: autoCreateResources.enabled Default Value: false	This flag enables the automation at the global level for all the resources that are configured under the list of service accounts. Note: Only the service account is configured as a resource.
Parameter name: serviceAccounts.create Default Value: false	This flag enables the service accounts to enable the automation at the resource level. You must enable both the autoCreateResources.enabled and serviceAccounts.create flags to automate the service account creation.

Table 2-5 (Cont.) Parameter Details

Parameter Details	Description
Parameter name: <code>serviceAccounts.accounts.type</code> Default Value: UDR	This field is not editable. The default value is always UDR.

Upgrade and Rollback

- During upgrade, if the service account automation is enabled, then a new service account is created as per the configurations. If the service account automation is disabled, then there will be no change in the existing behavior.
- If the new service account is created as per the automation, then during rollback, the service account is deleted and the previous version of the service account is restored.

Manual to Automation Upgrade

When upgrading from manual to automation with the LCM Based Automation feature enabled, it is recommended to always use the new service account name instead of the existing one. Along with the service account, a role and role binding is created with `<serviceaccount>-role` and `<serviceaccount>-rolebinding`. During the upgrade process, you should not delete the existing service account, which was created manually, as it will be required for rollback procedures.

Automation to Automation Upgrade

During automation (default creation of service account, role and role binding with release name) to automation (Helm automation) upgrade, it is recommended to use the same service account name instead of using a different service account name.

Observe

There are no metrics, KPIs, or alerts available for this feature.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.3 Traffic Segregation

This feature provides end-to-end traffic segregation to UDR based on traffic types. Within a Kubernetes cluster, traffic segregation can divide applications or workloads into distinct sections such as Operations Administration and Maintenance (OAM), Service Based Interface (SBI), Kubernetes control traffic, and so on. The Multus Container Network Interface (CNI) plugin for Kubernetes enables attaching multiple network interfaces to pods to help segregate traffic to and from UDR edge microservices like Ingress and Egress Gateway microservices.

This feature addresses the challenge of logically separating IP traffic of different profiles, which are typically handled through a single network (Kubernetes overlay). The new functionality ensures that critical networks are not cross-connected or sharing the same routes, thereby preventing network congestion.

With traffic segregation, operators can segregate traffic to external feeds and applications more effectively. Previously, all external traffic was routed through the same external network, but now, egress traffic from the UDR pods can be directed through non-default networks to third-party applications. This separation is achieved by leveraging cloud-native infrastructure and the load balancing algorithms in OCCNE.

The feature requires configuration of separate networks, Network Attachment Definitions (NADs), and the Cloud Native Load Balancer (CNLB). These configurations are required for facilitating Ingress/Egress traffic segregation, and optimizing load distribution in UDR.

Note

The Traffic Segregation feature is only available in UDR, SLF, and EIR if OCCNE is installed with CNLB.

Cloud Native Load Balancer (CNLB)

CNE provides CNLB for managing the ingress and egress network as an alternate to the existing LBVM, lb-controller, and egress-controller solutions. You can enable or disable this feature only during a fresh CNE installation. When this feature is enabled, CNE automatically uses CNLB to control ingress traffic. To manage the egress traffic, you must preconfigure the egress network details in the `cnlb.ini` file before installing CNE.

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

Network Attachment Definitions for CNLB

A Network Attachment Definition (NAD) is a resource used to set up a network attachment, in this case, a secondary network interface to a pod. UDR supports two types of CNLB NADs:

- 1. Ingress Network Attachment Definitions:** Ingress NADs are used to handle inbound traffic only. This traffic enters the CNLB application through an external interface service IP address and is routed internally using interfaces within CNLB networks.
 - Naming Convention:** `nf-<service_network_name>-int`
- 2. Egress Only Network Attachment Definitions:** Egress Only NADs enable outbound traffic only. An NF pod can initiate traffic and route it through a CNLB application, translating the source IP address to an external egress IP address. An egress NAD contains network information to create interfaces for NF pods and routes to external subnets.
 - Prerequisites:**
 - Ingress NADs are already created for the desired internal networks.
 - Destination (egress) subnet addresses are known beforehand and defined under the `cnlb.ini` file's `egress_dest` variable to generate NADs.
 - The use of an Egress NAD on a deployment can be combined with Ingress NADs to route traffic through specific CNLB apps.
 - Naming Convention:** `nf-<service_network_name>-egr`

Managing Ingress and Egress Traffic Segregation

Enable:

This feature is disabled by default. To enable this feature, you must configure the network attachment annotations in the `ocudr_custom_values.yaml` file.

Configuration:

For more information about Traffic Segregation configuration, see "Configuring Traffic Segregation" section in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no metrics, KPIs, or alerts available for this feature.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

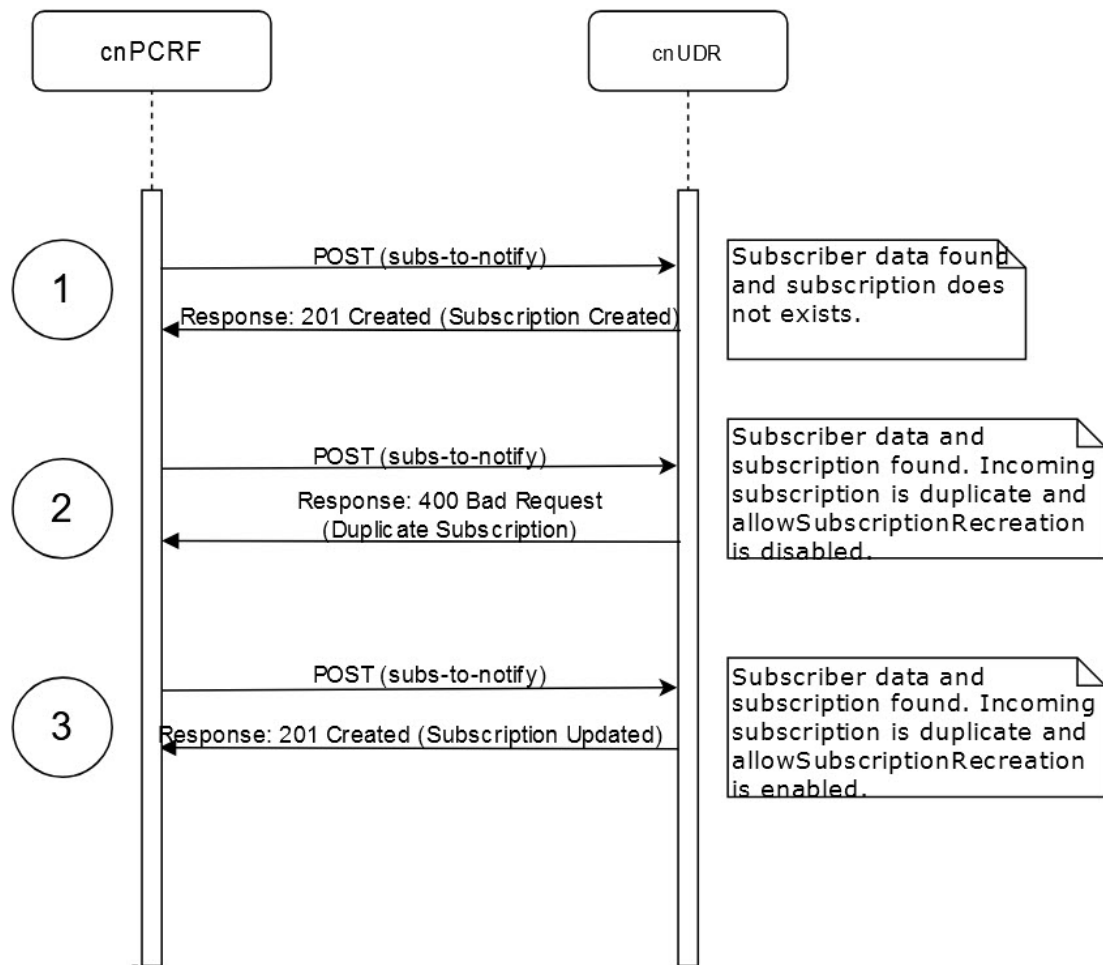
In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.4 Support for Post Operation for an Existing Subscription

When a DELETE request fails to remove the previous subscription, cnPCRF sends a POST request to cnUDR to create a new subscription, cnUDR would respond with a 400 Bad Request (Duplicate Subscription)' error since the subscription already existed in the database. With this feature, UDR supports the POST request from cnPCRF for subscription creation (subs-to-notify requests) that overwrites the existing subscription.

The feature is enabled by using the `allowSubscriptionRecreation` parameter. When the parameter is enabled the duplicate POST request sent is checked for the combination of `notificationUri` and `monitoredResourceUris` in the database. If the combination exists, then additional details, such as expiry time is updated in the database. If the parameter is disabled, then cnUDR responds with a 400 Bad Request (Duplicate Subscription) error.



- A combination of notificationUri and monitoredResourceUris parameters are used to check the duplicate POST requests.
- If the combination of notificationUri and monitoredResourceUris parameters are not found in the database, then a new subscription is created.
- If a combination is found in the database and the POST request is received multiple times with allowSubscriptionRecreation enabled, then the existing subscription is updated.

In the following example there is a unique combination of notificationUri and monitoredResourceUris, if any one of the parameters are changed, then a new subscription is created.

```

{
  "expires": "2099-04-23T18:25:43.511Z",
  "supportedFeatures": "f",
  "notificationUri": "http://notify-stub-service.ats-shriats:8080/policy-data/uri1/",
  "monitoredResourceUris": [
    "http://ocudr-ambassador.myudr.svc.cluster.local/nudr-dr/v2/policy-data/ues/msisdn-461772271137732/ue-policy-set",
    "http://ocudr-ambassador.myudr.svc.cluster.local/nudr-dr/v2/policy-data/ues/msisdn-461772271137732/sm-data/mk1",
    "http://ocudr-ambassador.myudr.svc.cluster.local/nudr-dr/v2/policy-data/ues/msisdn-461772271137732/sm-data"
  ]
}
  
```

```
    ]
}
```

Managing Post Operation for an Existing Subscription

Enable

You can enable or disable the feature by setting the Allow Subscription Recreation field to true or false in the [Data Repository Service](#) in the CNC Console.

Configure

You can configure this feature using REST APIs.

You can perform the PUT operation on the `/nudr-config/v1/udr.nudrservice.cfg/DR-SERVICE` to enable or disable the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "hikariPoolSize": 25,
  "defaultSlfGroupName": "DefaultGrp",
  "onDemandMigrationEnabled": true,
  "tracingEnabled": true,
  "defaultGroupIdEnabled": false,
  "subscriptionDataSubscriptionsOnly": false,
  "allowSubscriptionRecreation": true
}
```

Upgrade and Rollback of Post Operation for an Existing Subscription

The `allowSubscriptionRecreation` parameter must be added after upgrading from the previous releases and during rollback, the previous release configurations must be restored.

Observe

Metrics

The `udr_subscriptions_recreated` metric is used for this feature. For more information about the metric, see [nudr-dr-service Metrics](#).

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.5 Support for Kubernetes Resource

This section provides information about the Kubernetes resource features.

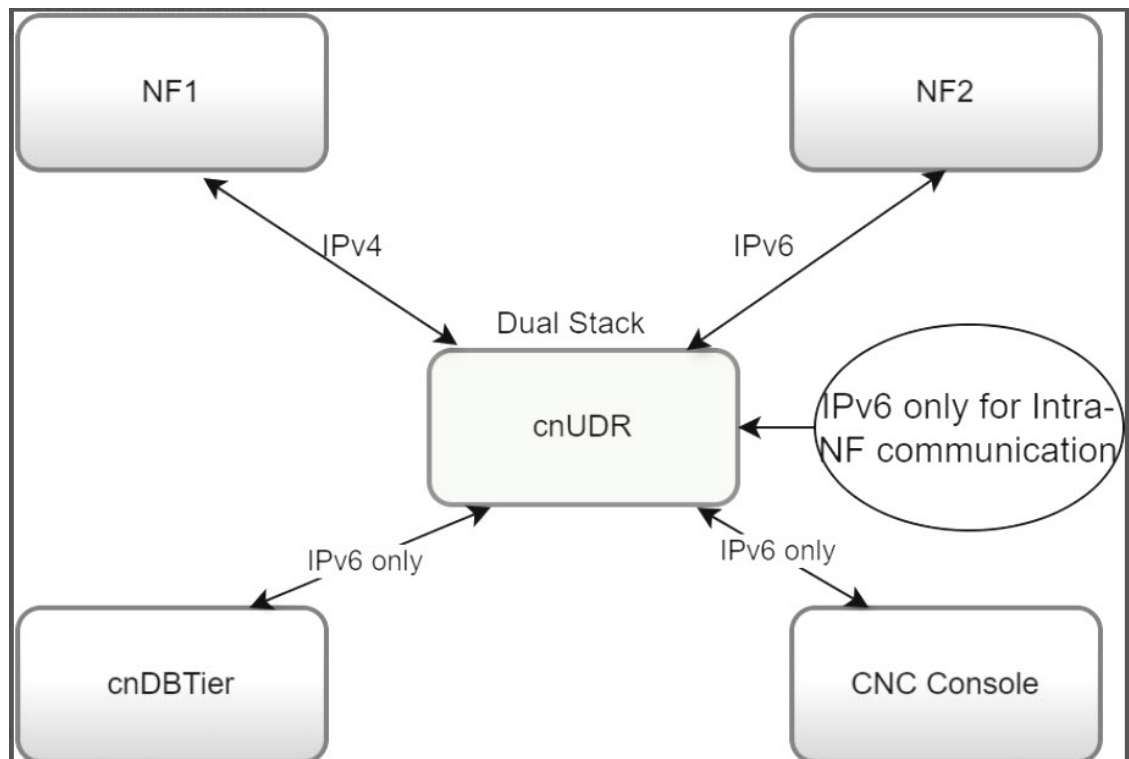
2.5.1 Support for Dual Stack

With this feature cnUDR can be deployed on a dual stack Kubernetes infrastructure. Using the dual stack mechanism, cnUDR establishes and accepts connections within pods and services in a Kubernetes cluster using IPv4 or IPv6. cnUDR communicates with the entities that support IPv4 and IPv6.

cnUDR supports:

- dual stack to communicate with the external entities. For example, NFs and provisioning client etc.
- only IPv6 communication within the cnUDR microservices.
- only IPv6 communication with cnDBTier, CNC Console, and Kubernetes APIs.

Figure 2-2 Dual Stack



The communication with the different entities is segregated as follows:

- External pods and services
- Internal pods and services

External Pods and Services

The expected behavior of external pods and services are as follows:

- For Ingress Gateway traffic:
 - Pods are dual stack and must be exposed to both IPv6 and IPv4 interfaces.

- Services are dual stack and must be exposed to both IPv6 and IPv4 services. The preferred IP must be IPv6 interface.
- For Egress Gateway traffic:
 - Pods are dual stack and must be exposed to both IPv6 and IPv4 interfaces. The preferred IP must be IPv6 interface. Use IPv4 interface depending on DNS resolution as follows:
 - * If DNS has only A and AAAA records, then you must use the IPv6 interface on the pod.
 - * If DNS has only A records, then you must use the IPv4 interface on the pod.

Internal Pods and Services

The expected behavior of internal pods and services are as follows:

- For Ingress Gateway traffic:
 - Pods are single stack and must be exposed to only IPv6 interfaces.
 - Services are single stack and must be exposed to only IPv6 services.
- For Egress Gateway traffic:
 - Pods are single stack and must be exposed to only IPv6 interfaces. The preferred IP must be IPv6 interface.

IP Address Allocation to Pods

IP address allocation to pods depends on the IP address preference set in the Kubernetes cluster. Pods do not have the privilege to choose an IP address. Consider the following example of a pod deployed in an IPv4 preferred infrastructure. Here, if the Kubernetes cluster has IPv4 preferred configuration, both IPv4 and IPv6 are allocated to the pod, but the primary IP address is IPv4. Example:

```
IP:                10.xxx.xxx.xxx
IPs:
  IP:              10.xxx.xxx.xxx
  IP:              fd00::1:cxxx:bxxx:8xxx:xxxx
```

IP Address Allocation to Services

IP address allocation to all the UDR services, depends on the `global.deploymentMode` Helm parameter configuration. This Helm parameter automatically configures IP Family Policy and IP Families attributes for the UDR services. For more information about `global.deploymentMode`, see the "Customizing UDR" section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

You can customize the IP address allocation to services based on the `global.deploymentMode` Helm parameter. Services route the traffic to the destination endpoints based on this configuration. If the `global.deploymentMode` Helm parameter is set to IPv4, then IPv4 is allocated to services, and services use IPv4 pod IPs to send the traffic to endpoints.

The following table describes how IP address allocation, IP Family Policy, and IP Families vary based on the `global.deploymentMode` Helm parameter configuration for services:

Table 2-6 IP Address Allocation

Infrastructure Preference	Application Preference (global.deploymentMode Helm Parameter)	IP Family Policy Attribute	IP Families Attribute	Pod IP	Service IP	Endpoints
IPv4 Preferred	IPv4	SingleStack	IPv4	IPv4,IPv6	IPv4	IPv4
IPv6 Preferred	IPv4	SingleStack	IPv4	IPv6,IPv4	IPv4	IPv4
IPv4 Preferred	IPv6	SingleStack	IPv6	IPv4,IPv6	IPv6	IPv6
IPv6 Preferred	IPv6	SingleStack	IPv6	IPv6,IPv4	IPv6	IPv6
IPv4 Preferred	IPv4_IPv6 (IPv4Preferred)	RequiredDualStack	IPv4 Preferred	IPv4,IPv6	IPv4,IPv6	IPv4
IPv6 Preferred	IPv4_IPv6 (IPv4Preferred)	RequiredDualStack	IPv4 Preferred	IPv6,IPv4	IPv6,IPv4	IPv4
IPv4 Preferred	IPv6_IPv4 (IPv6Preferred)	RequiredDualStack	IPv6 Preferred	IPv4,IPv6	IPv4,IPv6	IPv6
IPv6 Preferred	IPv6_IPv4 (IPv6Preferred)	RequiredDualStack	IPv6 Preferred	IPv6,IPv4	IPv6,IPv4	IPv6

Managing Dual Stack

Enable

You do not need to enable or disable this feature.

Configure

You can configure this feature using HELM.

- You can configure the feature by setting the `global.deploymentMode` parameter to indicate the deployment mode of the `cnUDR` in the `global` section of the `ocudr-custom-values.yaml` file. The default value is set as `ClusterPreferred`. The possible values are as follows:
 - IPv4
 - IPv6
 - IPv4_IPv6
 - IPv6_IPv4
 - ClusterPreferred

```
#Possible values : IPv4, IPv6, IPv4_IPv6, IPv6_IPv4, ClusterPreferred
global:
  deploymentMode: ClusterPreferred
```

`cnUDR` must be deployed on a dual stack Kubernetes infrastructure with IPv4 or IPv6 preferred CNE to use dual stack feature. The following table list deployment mode and its behavior.

Table 2-7 Deployment Mode

Deployment Mode	Description
IPv4	Only single stack and IPv4 preferred for all the services. If the deployment mode is set to IPv4, then all the communication between microservices are through IPv4. In this case, all the internal microservices and external services IP family policy are single stack and IP families will be IPv4.
IPv6	Only single stack and IPv6 preferred for all the services. If the deployment mode is set to IPv6, then all the communication between microservices are through IPv6. . In this case, all the internal microservices and external services IP family policy are single stack and IP families will be IPv6
IPv4_IPv6	Dual Stack and IPv4 preferred for all the services. If the deployment mode is set to IPv4_IPv6, then all the communication between microservices are through IPv4, since IPv4 is preferred. In this case, all the internal microservices IP family policy are single stack and IP families will be IPv4. The IP family policies of the external facing services are RequiredDualStack and IP families will be IPv4 and IPv6.
IPv6_IPv4	Dual Stack and IPv6 preferred for all the services. If the deployment mode is set to IPv6_IPv4, then all the communication between microservices are through IPv6, since IPv6 is preferred. In this case, all the internal microservices IP family policy are single stack and IP families will be IPv6. The IP family policies of the external facing services are RequiredDualStack and IP families will be IPv4 and IPv6.
ClusterPreferred	There will be no changes in the service file. IPs are assigned depending on the cluster. In this case, all the internal microservices IP family policy are single stack and IP families will be clustered IP address. The IP family policies of the external facing services are single stack and IP families will be clustered IP address.

Observe

There are no Metrics, KPIs, or Alerts available for this feature.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.5.2 Network Policies

Network Policies are an application-centric construct that allows you to specify how a pod can communicate with various network entities. Network Policies create pod-level rules to control communication between the cluster's pods and services. It determines which pods and services can access one another inside the cluster.

Previously, UDR could communicate with other namespaces, and pods of one namespace could communicate with other pods other namespaces without any restriction. In this release, namespace-level isolation is provided for the UDR pods, and communications are allowed between the UDR and pods outside the cluster. The network policies enforce access restrictions for all the applicable data flows except communication from Kubernetes node to pod for invoking container probe.

Managing Support for Network Policies

Enable

You can enable the feature by setting the `createNetworkPolicy` parameter to `true` in the global section of the `ocudr-custom values.yaml` file.

```
# Create network policy
createNetworkPolicy: true
```

Once the flag is set to `true`, you can deploy UDR by using `helm install` command and on successful installation, network policies are deployed.

You can disable the feature by setting the `createNetworkPolicy` parameter to `false` in the global section of the `ocudr-custom values.yaml` file. When the feature is disabled, and:

- If UDR is already deployed and upgrade is performed using the updated `ocudr-custom values.yaml` file, the network policies will not be present after the Helm upgrade.
- If it is a fresh UDR deployment and installation is performed using the updated `ocudr-custom values.yaml` file, the network policies will not be present after the Helm install.

The default network policy can be edited by using the `ocudr_custom_values.yaml` file. You must perform `helm upgrade` to apply the network policy rules.

Configure

You can configure this feature using Helm. For information about configuring Network Policies for UDR deployment, see *Oracle Communications Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no specific metrics and alerts required for the Network Policy feature.

2.5.3 Kubernetes Probes

One of the key feature that Kubernetes provides is high availability. This is achieved by the smallest deployment model called Pods. The health check of these Pods are performed by Kubernetes Probes.

There are three types of probes:

- **Liveness Probe:** Indicates if the container is operating. If the container is operating, no action is taken. If not, the kubelet kills and restarts the container.
- **Readiness Probe:** Indicates whether the application running in the container is ready to accept requests. If the application is ready, services matching the pod are allowed to send traffic to it. If not, the endpoints controller removes the pod from all matching Kubernetes Services.
- **Startup Probe:** Indicates whether the application running in the container has started. If the application is started, other probes start functioning. If not, the kubelet kills and restarts the container.

2.6 Suppress Notification

Earlier, the Cloud Native Unified Data Repository (cnUDR) was sending notifications to Cloud Native Policy and Charging Rules Function (cnPCRF) Ingress Gateway endpoint when sm-data PATCH or subs-to-notify POST request was received from cnPCRF Egress Gateway endpoint. cnUDR was unable to determine if the update operations (PUT, PATCH, or DELETE) received were from the same cnPCRF or different cnPCRF. cnUDR requires a mechanism to validate the source of the update operations to regulate the notifications.

Using the Suppress Notification feature, cnUDR can store the User-Agent header received in the POST request from cnPCRF in the subscription table. cnUDR compares the User-Agent header received during an update operation from cnPCRF with the stored User-Agent header. If the User-Agent header matches, the notification is suppressed. If the User-Agent header does not match or if there is no User-Agent header in the update request, the notification is sent to cnPCRF.

This feature enables cnUDR to identify the unique cnPCRF instance that has sent the update operations and suppress notifications to the same cnPCRF instance if the User-Agent header matches. This is achieved by using the User-Agent header in the HTTP requests received on cnUDR. The User-Agent header feature must be enabled from cnPCRF Egress Gateway. For more information about the User-Agent header, see *"Support for User-Agent header"* in *Oracle Communications Cloud Native Core Policy User Guide*.

The User-Agent header feature sends cnPCRF NFInstanceID and NF FQDN for every request toward cnUDR. When cnUDR receives sm-data PATCH or subs-to-notify POST from cnPCRF, cnUDR stores the User-Agent header along with the corresponding subscription in the subscription table. When cnUDR receives another update request (PATCH, PUT, or DELETE), the received User-Agent header is compared with the User-Agent header stored in the subscription table. The notification is suppressed if the User-Agent matches.

The Suppress Notification feature is applicable only for signaling requests from cnPCRF, not for provisioning requests

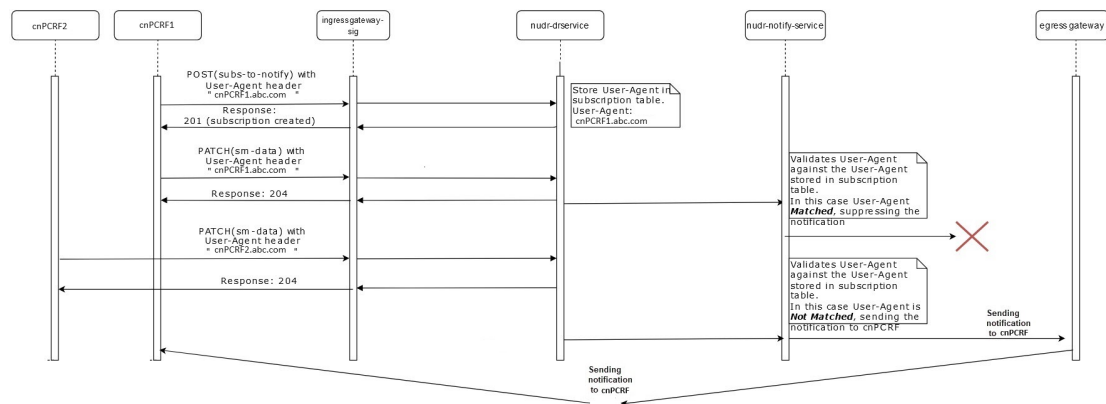
Prerequisites

You must enable the User-Agent header feature from cnPCRF. For more information about the feature, see *Support for User-Agent Header* section in *Oracle Communications Cloud Native Core Policy User Guide*.

Signaling Requests

The following call flow explains how the notifications is suppressed if the suppress notification feature is enabled.

Figure 2-3 Signaling Requests

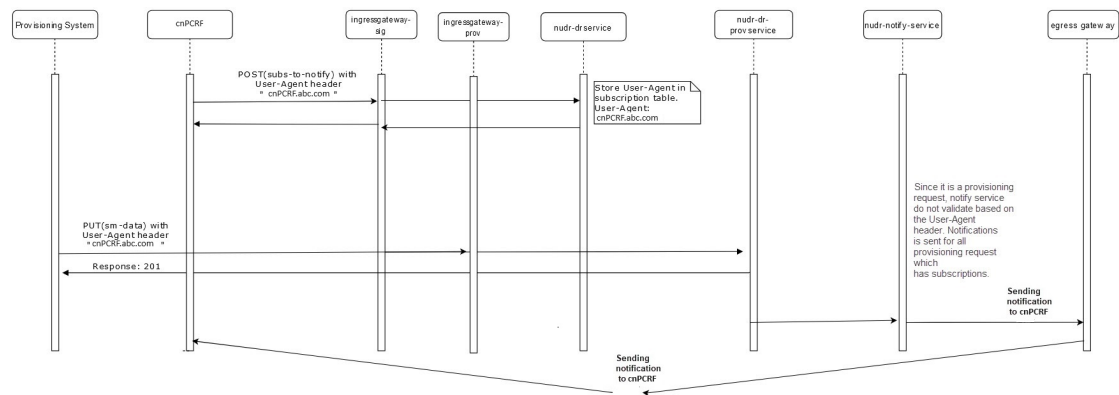


1. When cnPCRF1 sends a POST (subs-to-notify) request to cnUDR, cnUDR collects the User-Agent header (Example, cnPCRF1.abc.com) from the request and saves it in the subscription table against the subscriber.
2. If cnPCRF1 sends another PATCH request to cnUDR with the same User-Agent header (Example, cnPCRF1.abc.com), then notify-service validates the User-Agent header with the saved User-Agent header in the subscription table. Since the User-Agent header matches in this case the notification is suppressed.
3. If cnPCRF2 sends a PATCH request to cnUDR with a different User-Agent header (Example, cnPCRF2.abc.com), notify-service validates the User-Agent header with the saved User-Agent header in the subscription table. In this case, the User-Agent header does not match and the notification is sent.

Provisioning Requests

For the provisioning requests, the notifications are sent for all the update operations irrespective of the feature is enabled or disabled. The following call flow explains that how the notifications are sent for provisioning requests.

Figure 2-4 Provisioning Requests



1. When cnPCRF sends a POST (subs-to-notify) to cnUDR, cnUDR collects the User-Agent header (Example, cnPCRF.abc.com) from the request and saves it in the subscription table against the subscriber.
2. If an update request from the provisioning system with the same User-Agent header (Example, cnPCRF.abc.com) is sent from POST subs-to-notify, it is not validated and notification is triggered.

Multiple User-Agents Header

When the feature is enabled and if multiple User-Agents header are received in the request only the first occurrence is considered and saved in the subscription table.

For example, if two User-Agents header are received in the request, such as User-Agent: PCF-123 and User-Agent: NRF-123. Only User-Agent; PCF-123 is considered and saved in the subscription table.

Managing Suppress Notification

Enable

You can enable the suppress notification feature using Helm Configuration:

- You can enable or disable the feature by setting the `suppressNotificationEnabled` parameter to true or false in the global section of the `ocudr-custom-values.yaml` file. When the flag is set to true, the notification is suppressed if the User-Agent header matches. The default value is set as true. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure this feature using REST API and the CNC Console.

- **Configure using CNC Console:**
You can disable or enable the feature by setting the `suppressNotificationEnabled` parameter to true or false in the [Global Configurations](#) in the CNC Console.
- **Configure using REST API:**
You can perform the PUT operation on the `http://10.75.229.45:30015/nudr-config/v1/udr.global.cfg/GLOBAL` to enable or disable the feature. For more

information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "dbServiceName": "mysql-connectivity-service.occne-ndb",
  "udrServices": "All",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "sbiCorrelationInfoEnable": false,
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "vsaDefaultBillingDay": 1,
  "subscriberActivityEnabled": false,
  "addDefaultBillingDay": true,
  "enableControlledShutdown": true,
  "keyType": "imsi",
  "keyRange": "2013020001-2013020002",
  "nfType": "UDR",
  "nfFqdn": "udr001.oracle.com",
  "suppressNotificationEnabled": true,
  "configClientConnectTimeout": 1000,
  "configClientReadTimeout": 1000,
  "subscriberIdentifiers": {
    "msisdn": [],
    "imsi": [],
    "nai": [],
    "extid": []
  }
}
```

Upgrade and Rollback of Suppress Notification Feature

Suppress notification feature works for subscriptions created on 24.2.0. release version only. Suppress notification feature does not work as expected if an upgrade or rollback is performed on the subscriptions created on a different release version.

For example:

- **Rollback scenario:** If the subscriptions is created on 24.2.0 release versions, then if a rollback to 24.1.x release or older release version is performed, the notifications are not generated or suppressed as expected.
- **Upgrade scenario:** If the subscriptions is created on 24.1.x release or older release versions, then if an upgrade to 24.2.0 release version is performed, the notifications are not generated or suppressed as expected.

It is recommended to create a new subscriptions on upgraded or rollback release versions for the suppress notification feature to work as expected.

Observe

Metrics

The following metrics are used for this feature:

- `suppress_user_agent_not_found`
- `nudr_notif_suppress_notifications`

Alerts:

The following alert is generated for this feature:

- [UserAgentHeaderNotFoundMorethan10PercentRequest](#)

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.7 Support for Ingress Gateway and Egress Gateway APIs in CNC Console

Earlier, the configuration for common service APIs was supported only using REST. With the implementation of this feature, UDR supports the configuration of Ingress Gateway and Egress Gateway parameters using the CNC Console.

For more information about the common service APIs, see "Configuration APIs for Common Services" section in the *Oracle Communication Cloud Native Core, Unified Data Repository REST Specification Guide*.

Managing common service APIs in the CNC Console

Enable

This feature is enabled automatically along with the UDR instance deployment.

Configure

You can configure the common services APIs in the CNC Console. For more information, see [Egress Gateway](#) and [Ingress Gateway](#) sections.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.8 Support for Automated Certificate Lifecycle Management

UDR uses secure protocols, such as HTTPS and Secure Socket Layer (SSL) or Transport Layer Security (TLS), to establish and manage secure connections. This is achieved using Public and Private Keys and the presence of trusted authorities such as Certificate Authorities (CA), which create and issue certificates. These certificates have validity. You must renew these certificates before they expire. These certificates can be revoked when the CA or its keys are compromised

Starting with UDR 24.2.x, you can integrate UDR with Oracle Communications Cloud Native Core, Certificate Management (OCCM) to support automation of certificate lifecycle management. OCCM manages TLS certificates stored in Kubernetes secrets by integrating with Certificate Authority (CA) using the Certificate Management Protocol Version 2 (CMPv2) protocol in the Kubernetes secret. OCCM obtains and signs TLS certificates within the UDR namespace. For more information about OCCM, see *Oracle Communications Cloud Native Core, Certificate Management User Guide*.

Figure 2-5 Support for OCCM



The above diagram indicates that OCCM writes the keys to the certificates and UDR reads these keys to establish a TLS connection with other NFs.

OCCM can automatically manage the following TLS certificates:

- 5G Service Based Architecture (SBA) client TLS certificates
- 5G SBA server TLS certificates
- Message Feed TLS certificates

Install Guide Considerations

- Upgrade: When UDR is deployed with OCCM, follow the specific upgrade procedure. For information about the upgrade strategy, see "Upgrading UDR" in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- Rollback: For more information on migrating the secrets from UDR to OCCM and removal of Kubernetes secrets from the yaml file, see "Postupgrade Task" in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Configure

There are no additional configuration changes required at UDR.

Observe

This feature uses the existing metrics:

- `oc_egressgateway_connection_failure_total`
- `oc_ingressgateway_connection_failure_total`

Maintain

If you encounter any OCCM-specific alerts, see the "OCCM Alerts" section in *Oracle Communications Cloud Native Core, Certificate Management User Guide*.

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.9 Support for TLS

UDR use Hypertext Transfer Protocol Secure (HTTPS) to establish secured connections with consumer NFs and producer NFs, respectively. These communication protocols are encrypted using Transport Layer Security (TLS).

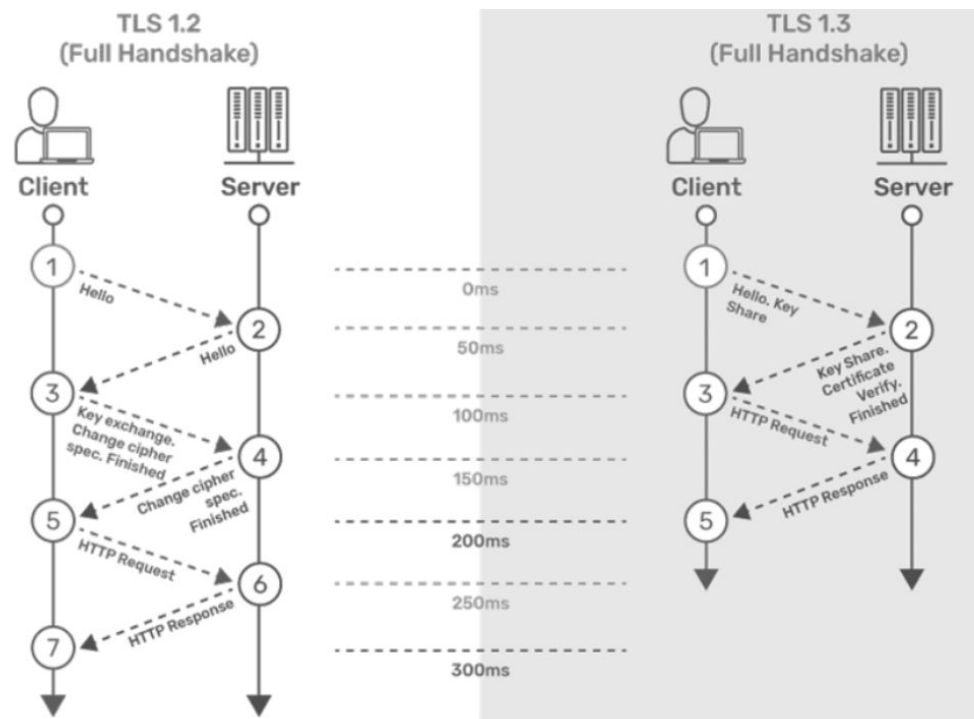
TLS comprises the following components:

- **Handshake Protocol:** Exchanges the security parameters of a connection. Handshake messages are supplied to the TLS record layer.
- **Record Protocol:** Receives the messages to be transmitted, fragments the data into multiple blocks, secures the records, and then transmits the result. Received data is delivered to higher-level peers.

This feature enables the support for TLS 1.3 to all consumer NFs, producer NFs, the Data Director, SBI Interfaces, and any interfaces previously supporting TLS 1.2. Support for TLS 1.2 will remain available.

TLS Handshake

This section describes the differences between TLS 1.3 and TLS 1.2 and the advantages of TLS 1.3 over TLS 1.2 and earlier versions.



TLS 1.2

Step 1: The connection or handshake starts when the client sends a "client hello" message to the server. This message consists of cryptographic information such as supported protocols and cipher suites. It also contains a random value or random byte string.

Step 2: To respond to the "client hello" message, the server sends a "server hello" message. This message contains the CipherSuite that the server has selected from the options provided by the client. The server also sends its certificate, along with the session ID and another random value.

Step 3: The client verifies the certificate sent by the server. When the verification is complete, it sends a byte string encrypted using the public key of the server certificate.

Step 4: When the server receives the secret, both the client and server generate a master key along with session keys (ephemeral keys). These session keys are used for symmetrically encrypting the data.

Step 5: The client sends an "HTTP Request" message to the server to enable the server to transition to symmetric encryption using the session keys.

Step 6: To respond to the client's "HTTP Request" message, the server does the same and switches its security state to symmetric encryption. The server concludes the handshake by sending an HTTP response.

Step 7: The client-server handshake is completed in two round trips.

TLS 1.3

Step 1: The connection or handshake begins when the client sends a "client hello" message to the server, which includes the list of supported cipher suites. The client also sends its key share for that particular key agreement protocol.

Step 2: To respond to the "client hello" message, the server sends the key agreement protocol that it has chosen. The "Server Hello" message includes the server key share, server certificate, and the "Server Finished" message.

Step 3: The client verifies the server certificate, generates keys since it has the server's key share, and then sends the "Client Finished" message along with an HTTP request.

Step 4: The server completes the handshake by sending an HTTP response.

The following digital signature algorithms are supported in TLS handshake:

Table 2-8 Digital Signature Algorithms

Algorithm	Key Size (Bits)	Elliptic Curve (EC)
RS256 (RSA)	2048	NA
	4096 This is the recommended value.	NA
ES256 (ECDSA)	NA	SECP384r1 This is the recommended value.

Comparison Between TLS 1.2 and TLS 1.3

The following table provides comparison of TLS 1.2 with TLS 1.3:

Table 2-9 Comparison of TLS 1.2 and TLS 1.3

Feature	TLS 1.2	TLS 1.3
TLS Handshake	- The initial handshake was carried out in clear text. - A typical handshake in TLS v1.2 involves the exchange of 5 to 7 packets.	- The initial handshake is carried out along with the key share. - A typical handshake IN TLS v1.3 involves the exchange of up to 3 packets.
Cipher Suites	- Less secure Cipher suites. - Use SHA-256 and SHA-384 hashing - 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	- More secure Cipher suites. - Apart from all the ciphers supported for TLS 1.2 , the following additional ciphers are supported for only TLS 1.3 : - TLS_AES_128_GCM_SHA256 - TLS_AES_256_GCM_SHA384 - TLS_CHACHA20_POLY1305_SHA256
Round-Trip Time (RTT)	This has a high RTT during the TLS handshake.	This has low RTT during the TLS handshake.
Perfect Forward Secrecy (PFS)	This doesn't support PFS.	TLS V1.3 supports PFS. PFS ensures that each session key is completely independent of long-term private keys, which are keys that are used for an extended period to decrypt encrypted data.
Privacy	This is less secure, as the ciphers used are weak.	This is more secure, as the ciphers used are strong.

Table 2-9 (Cont.) Comparison of TLS 1.2 and TLS 1.3

Feature	TLS 1.2	TLS 1.3
Performance	This has high latency and a less responsive connection.	This has low latency and a more responsive connection.

Note

- UDR does not prioritize cipher suites on the basis of priorities. To select cipher on the basis of priorities, you must write the cipher suites in the decreasing order of priority.
- UDR does not prioritize supported groups on the basis of priorities. To select supported group on the basis of priorities, you must write the supported group values in the decreasing order of priority.
- If you want to provide values for the signature_algorithms extension using the `clientSignatureSchemes` parameter, the following comma-separated values must be provided to deploy the pods:
 - `rsa_pkcs1_sha512`
 - `rsa_pkcs1_sha384`
 - `rsa_pkcs1_sha256`
- The mandatory extensions as listed in RFC 8446 cannot be disabled on the client or server side. The following is the list of the extensions that cannot be disabled:
 - `supported_versions`
 - `key_share`
 - `supported_groups`
 - `signature_algorithms`
 - `pre_shared_key`

Advantages of TLS 1.3

TLS 1.3 handshake offers the following improvements over earlier versions:

- In TLS 1.3, all handshake messages after the ServerHello are encrypted.
- TLS 1.3 improves efficiency in the handshake process by requiring fewer round trips than TLS 1.2. It also uses cryptographic algorithms that are faster.
- TLS 1.3 has better security than TLS 1.2. It addresses known vulnerabilities in the handshake process.
- TLS 1.3 got rid of data compression.

The following table describes the TLS versions supported in the client and server side. The last column indicates which version is used.

Table 2-10 TLS Combinations

Client	Server	TLS Version Used
TLS1.2+1.3	TLS1.2+1.3	TLSv1.3
TLSv1.3	TLSv1.3	TLSv1.3
TLSv1.3	TLSv1.2+1.3	TLSv1.3
TLSv1.2+1.3	TLSv1.3	TLSv1.3
TLSv1.2	TLSv1.2+1.3	TLSv1.2
TLSv1.2+1.3	TLSv1.2	TLSv1.2
TLSv1.3	TLSv1.2	FATAL sent by server
TLSv1.2	TLSv1.3	FATAL sent by server

Note

- If Egress Gateway is deployed with both the versions of TLS (TLSv1.2 and TLSv1.3), then Egress Gateway as client sends both versions of TLS in the client hello message during the handshake and the server needs to decide which version to be used.
- If Ingress Gateway is deployed with both the version of TLS (TLSv1.2 and TLSv1.3), then Ingress Gateway as the server uses the TLS version received from the client in the server hello message during the handshake.

Managing Support for TLS v1.2 and 1.3**Enable**

This feature can be enabled or disabled at the time of UDR deployment for both Ingress and Egress traffic. For more information on enabling `initssl`, `enableIncomingHttps`, and `outgoingHttps` flags, see the *"Enabling HTTPS on Ingress Gateway"* and *Enabling HTTPS on Egress Gateway* sections in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Configure

Generate HTTPS certificates for both the ingress and egress gateways. Ensure that the certificates are correctly configured for secure communication. After generating the certificates, create a Kubernetes secret for each gateway (egress and ingress). Then, configure these secrets to be used by the respective gateways.

You can configure this feature using Helm parameters. The following parameters in the Ingress Gateway and Egress Gateway microservices must be customized to support TLS 1.2 or TLS 1.3.

The following parameters are required to support TLS 1.2:

- `service.ssl.tlsVersion` indicates the TLS version.
- `cipherSuites` indicates supported cipher suites.
- `allowedCipherSuites` indicates allowed Ciphers suites.

The following parameters are required to support TLS 1.3:

- `service.ssl.tlsVersion` indicates the TLS version.

- `cipherSuites` indicates the supported cipher suites.
- `allowedCipherSuites` indicates the allowed cipher suites.
- `clientDisabledExtension` is used to disable the extension sent by messages originated by clients during the TLS handshake with the server.
- `serverDisabledExtension` is used to disable the extension sent by messages originated by servers during the TLS handshake with the client.
- `tlsNamedGroups` is used to provide a list of values sent in the `supported_groups` extension. These are comma-separated values.
- `clientSignatureSchemes` is used to provide a list of values sent in the `signature_algorithms` extension. These are comma-separated values.

For more information about configuring the values of the above-mentioned parameters, see "ocudr-ingressgateway-sig Microservice Parameters", "ocudr-ingressgateway-prov Microservice Parameters", and "ocudr-egressgateway Microservice Parameters" sections in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*

Observe

Metrics

The following metrics are available for this feature:

- `oc_ingressgateway_incoming_tls_connections`
- `oc_egressgateway_outgoing_tls_connections`
- `security_cert_x509_expiration_seconds`

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.10 Support for Provisioning Logs

The provisioning log feature enhances the provisioning logging functionality by enabling a logging type that is independent of the logging level. This feature helps in collecting the subscriber provisioning logs by enabling the provisioning logging mechanism. It also enables the operators to log the 5G subscriber provisioning requests for create, update, and delete operations.

Provisioning logging mechanism is defined in the following format with a key value pair:

```
{"timeStamp":"","logType":"","correlationId":"","nfInstanceId":"","method":"","request/response":""}
```

- `timeStamp`: It is the request and response time provided in Tue, 07 Jun 2021 01:11:22 GMT format.

- logType: It is the differentiator that is used to search in OpenSearch.
- correlationId: It identifies the provisioning transaction
- nflInstanceId: Identifies the site to which the logs belong.
- method: It is the HTTP method.
- request/response: It is the JSON request payload or JSON response payload.

Sample logging of provisioning request is as follows:

Request :

```
{
  "instant": {
    "epochSecond": 1708691773, "nanoOfSecond": 619264712, "thread": "XNIO-1
task-2", "level": "INFO", "loggerName": "ocudr.udr.services.common.model.UdrDataUr
i", "message": "Request received: http://10.75.229.173:31331/nudr-dr-prov/v2/
policy-data/msisdn-33123654862/sm-data/vsa {\\r\\n    \\subscriber\\":
{\\r\\n    \\IMSI\\": [\\\"302370123456789\\\"], \\r\\n    \\ACCOUNTID\\":
[\\\"178322212122\\\"], \\r\\n    \\MSISDN\\": [\\\"33123654862\\\"], \\r\\n
\\\"BillingDay\\": 6, \\r\\n    \\Entitlement\\": [\\\"DayPass\\\", \\\"DayPassPlus\\"]
\\r\\n    }, \\r\\n    \\usage\\": {\\r\\n    \\quota\\": [{\\r\\n
\\\"totalVolume\\": \\\"100\\\", \\r\\n    \\Type\\":
\\\"quota\\\", \\r\\n    \\name\\": \\\"Weekend\\\", \\r\\n
\\\"cid\\": \\\"9223372036854775808\\\", \\r\\n    \\nextResetTime\\":
\\\"2014-01-10T02:00:00Z\\\", \\r\\n    \\QuotaState\\":
\\\"active\\\", \\r\\n    }, \\r\\n    {\\r\\n
\\\"totalVolume\\": \\\"100\\\", \\r\\n    \\Type\\":
\\\"quota\\\", \\r\\n    \\name\\": \\\"Evenings\\\", \\r\\n
\\\"cid\\": \\\"9223372036854775809\\\", \\r\\n    \\nextResetTime\\":
\\\"2014-02-01T00:00:00Z\\\", \\r\\n    \\QuotaState\\":
\\\"active\\\", \\r\\n    }, \\r\\n    }, \\r\\n    \\version\\":
3, \\r\\n    }, \\r\\n    \\definition\\": {\\r\\n    \\SequenceNumber\\":
1, \\r\\n    \\version\\": 1, \\r\\n    \\DynamicQuota\\": [{\\r\\n
\\\"InstanceId\\": \\\"15678\\\", \\r\\n    \\Priority\\": \\\"4\\\", \\r\\n
\\\"InitialInputVolume\\": \\\"15000\\\", \\r\\n
\\\"InterimReportingInterval\\": \\\"1000\\\", \\r\\n    \\Duration\\":
\\\"100\\\", \\r\\n    \\InitialOutputVolume\\": \\\"5000\\\", \\r\\n
\\\"Type\\": \\\"pass\\\", \\r\\n    \\activationdatetime\\":
\\\"2015-03-09T11:20:32\\\", \\r\\n    \\expirationdatetime\\":
\\\"2015-04-09T11:20:32\\\", \\r\\n    \\InitialTotalVolume\\":
\\\"20000\\\", \\r\\n    \\name\\": \\\"DynamicAggregateLimit\\\", \\r\\n
\\\"InitialServiceSpecific\\": \\\"9\\\", \\r\\n    \\InitialTime\\":
\\\"135\\\", \\r\\n    }, \\r\\n    }, \\r\\n    \\state\\": {\\r\\n    \\property\\":
[{\\r\\n    \\name\\": \\\"mcc\\\", \\r\\n    \\value\\":
\\\"123\\\", \\r\\n    }, {\\r\\n    \\name\\": \\\"expire\\\", \\r\\n
\\\"value\\": \\\"201402-09T11:20:32\\\", \\r\\n    }, {\\r\\n
\\\"name\\": \\\"version\\":
1, \\r\\n    }
\\r\\n}], \\r\\n    \\endOfBatch\\": false, "loggerFqn": "org.apache.logging.slf4j.Log4jLogger",
"threadId": 191, "threadPriority": 5, "messageTimestamp": "2024-02-23T12:36:13.619+
0000", "application": "ocudr", "engVersion": "24.1.0-
beta.54", "mktgVersion": "23.4.0.0.0", "microservice": "ocudr-nudr-dr-
provservice", "vendor": "Oracle", "subscriberId": "msisdn-33123654862", "resourceId
": "nudr-dr-prov/v2/policy-data/sm-data/
vsa", "ocLogId": "1708691773617_191_ocudr-nudr-dr-provservice-5b58c78644-
xd6gh", "requestType": "POST"}
```

Response:

```
{
  "instant": {
    "epochSecond": 1708691773,
    "nanoOfSecond": 641078272,
    "thread": "XNIO-1 task-2",
    "level": "INFO",
    "loggerName": "ocudr.udr.services.common.services.VsaDbH andler",
    "message": "creating smPolicySnssaiData using default values",
    "endOfBatch": false,
    "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger",
    "threadId": 191,
    "threadPriority": 5,
    "messageTimestamp": "2024-02-23T12:36:13.641+0000",
    "application": "ocudr",
    "engVersion": "24.1.0-beta.54",
    "mktgVersion": "23.4.0.0.0",
    "microservice": "ocudr-nudr-dr-provservice",
    "vendor": "Oracle",
    "subscriberId": "msisdn-33123654862",
    "resourceId": "nudr-dr-prov/v2/policy-data/sm-data/vsa",
    "ocLogId": "1708691773617_191_ocudr-nudr-dr-provservice-5b58c78644-xd6gh",
    "requestType": "POST"
  }
}
```

Sample request for POST, PUT, PATCH, and DELETE are as follows:

- **POST:**

- Success request and response:

```
{
  "timeStamp": "Mon, 22 Jan 2024 10:52:05 UTC",
  "logType": "OCUDR-PROV-LOGS",
  "correlationId": "1708686574409-61110498-0699-420d-baae-770fb5623a10",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "method": "POST",
  "request": {
    "subscriber": {
      "MSISDN": "33123654863",
      "IMSI": "302370123456782",
      "ACCOUNTID": "178322212123",
      "BillingDay": 6,
      "Entitlement": "DayPass",
      "DayPassPlus": {}
    },
    "usage": {
      "quota": {
        "totalVolume": "100",
        "Type": "quota",
        "name": "Weekend",
        "nextResetTime": "2014-01-10T02:00:00Z",
        "cid": "9223372036854775808",
        "QuotaState": "active"
      },
      "totalVolume": "100",
      "Type": "quota",
      "name": "Evenings",
      "nextResetTime": "2014-02-01T00:00:00Z",
      "cid": "9223372036854775809",
      "QuotaState": "active"
    }
  },
  "version": 3,
  "definition": {
    "DynamicQuota": {
      "InstanceId": "15678",
      "Priority": "4",
      "InitialInputVolume": "15000",
      "InterimReportingInterval": "1000",
      "Duration": "100",
      "InitialOutputVolume": "5000",
      "Type": "pass",
      "activationdatetime": "2015-03-09T11:20:32",
      "expirationdatetime": "2015-04-09T11:20:32",
      "InitialTotalVolume": "20000",
      "name": "DynamicAggregateLimit",
      "InitialServiceSpecific": "9",
      "InitialTime": "135"
    }
  },
  "sequenceNumber": 1,
  "version": 1,
  "state": {
    "property": {
      "name": "mcc",
      "value": "123"
    },
    "name": "expire",
    "value": "201402-09T11:20:32"
  },
  "version": 1
}
{
  "timeStamp": "Mon, 22 Jan 2024 10:52:05 UTC",
  "logType": "OCUDR-PROV-LOGS",
  "correlationId": "1708686574409-61110498-0699-420d-baae-770fb5623a10",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "method": "POST",
  "response": "201 CREATED"
}
```

- Failure request and response:

```
{
  "timeStamp": "Mon, 22 Jan 2024 10:58:07 UTC",
  "logType": "OCUDR-PROV-LOGS",
  "correlationId": "1708686574419-61110498-0699-420d-baae-770fb5623a11",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "method": "POST",
  "request": {
    "subscriber": {
      "MSISDN": "33123654863",
      "IMSI": "302370123456782",
      "ACCOUNTID": "178322212123",
      "BillingDay": 6,
      "Entitlement": "DayPass",
      "DayPassPlus": {}
    },
    "usage": {
      "quota": {
        "totalVolume": "100",
        "Type": "quota",
        "name": "Weekend",
        "nextResetTime": "2014-01-10T02:00:00Z",
        "cid": "9223372036854775808",
        "QuotaState": "active"
      },
      "totalVolume": "100",
      "Type": "quota",
      "name": "Evenings",
      "nextResetTime": "2014-02-01T00:00:00Z",
      "cid": "9223372036854775809",
      "QuotaState": "active"
    }
  },
  "version": 3,
  "definition": {
    "DynamicQuota": {
      "InstanceId": "15678",
      "Priority": "4",
      "InitialInputVolume": "15000",
      "InterimReportingInterval": "1000",
      "Duration": "100",
      "InitialOutputVolume": "5000",
      "Type": "pass",
      "activationdatetime": "2015-03-09T11:20:32",
      "expirationdatetime": "2015-04-09T11:20:32",
      "InitialTotalVolume": "20000",
      "name": "DynamicAggregateLimit",
      "InitialServiceSpecific": "9",
      "InitialTime": "135"
    }
  },
  "sequenceNumber": 1,
  "version": 1,
  "state": {
    "property": {
      "name": "mcc",
      "value": "123"
    },
    "name": "expire",
    "value": "201402-09T11:20:32"
  },
  "version": 1
}
```

```
014-01-10T02:00:00Z", "cid": "9223372036854775808", "QuotaState": "active"},
{"totalVolume": "100", "Type": "quota", "name": "Evenings", "nextResetTime": "2
014-02-01T00:00:00Z", "cid": "9223372036854775809", "QuotaState": "active"}]
, "version": 3}, "definition": {"DynamicQuota":
[{"InstanceId": "15678", "Priority": "4", "InitialInputVolume": "15000", "Inte
rimReportingInterval": "1000", "Duration": "100", "InitialOutputVolume": "500
0", "Type": "pass", "activationdatetime": "2015-03-09T11:20:32", "expirationd
atetime": "2015-04-09T11:20:32", "InitialTotalVolume": "20000", "name": "Dyna
micAggregateLimit", "InitialServiceSpecific": "9", "InitialTime": "135"}], "S
equenceNumber": 1, "version": 1, "state": {"property":
[{"name": "mcc", "value": "123"},
{"name": "expire", "value": "201402-09T11:20:32"}], "version": 1}}}]
{"Timestamp": "Mon, 22 Jan 2024 10:58:07 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1708686574419-61110498-0699-420d-
baae-770fb5623a11", "nfInstanceId": "5a7bd676-ceed-44bb-95e0-
f6a55a328b03", "method": "POST", "response": "400 BAD_REQUEST"}
```

- **PUT:**

- Success request and response:

```
{"timeStamp": "Mon, 22 Jan 2024 10:53:04 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1705920784991-61110498-0699-420d-
baae-770fb5623a12", "nfInstanceId": "5a7bd676-ceed-44bb-95e0-
f6a55a328b03", "method": "PUT", "request": {"subscriber": {"MSISDN":
["33123654863"], "IMSI": ["302370123456782"], "ACCOUNTID":
["178322212123"], "BillingDay": 6, "Entitlement":
["DayPass", "DayPassPlus"]}, "usage": {"quota":
[{"totalVolume": "100", "Type": "quota", "name": "Weekend", "nextResetTime": "2
014-01-10T02:00:00Z", "cid": "9223372036854775808", "QuotaState": "active"},
{"totalVolume": "100", "Type": "quota", "name": "Evenings", "nextResetTime": "2
014-02-01T00:00:00Z", "cid": "9223372036854775809", "QuotaState": "active"}]
, "version": 3}, "definition": {"DynamicQuota":
[{"InstanceId": "15678", "Priority": "4", "InitialInputVolume": "15000", "Inte
rimReportingInterval": "1000", "Duration": "100", "InitialOutputVolume": "500
0", "Type": "pass", "activationdatetime": "2015-03-09T11:20:32", "expirationd
atetime": "2015-04-09T11:20:32", "InitialTotalVolume": "20000", "name": "Dyna
micAggregateLimit", "InitialServiceSpecific": "9", "InitialTime": "135"}], "S
equenceNumber": 1, "version": 1, "state": {"property":
[{"name": "mcc", "value": "123"},
{"name": "expire", "value": "201402-09T11:20:32"}], "version": 1}}}]
{"Timestamp": "Mon, 22 Jan 2024 10:53:05 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1705920784991-61110498-0699-420d-
baae-770fb5623a12", "nfInstanceId": "5a7bd676-ceed-44bb-95e0-
f6a55a328b03", "method": "PUT", "response": "201 CREATED"}
```

- Failure request and response:

```
{"timeStamp": "Mon, 22 Jan 2024 11:00:54 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1705921254488-61110498-0699-420d-
baae-770fb5623a13", "nfInstanceId": "5a7bd676-ceed-44bb-95e0-
f6a55a328b03", "method": "PUT", "request": {"subscriber": {"MSISDN":
["33123654863"], "IMSI": ["302370123456782"], "ACCOUNTID":
["178322212123"], "BillingDay": 6, "Entitlement":
["DayPass", "DayPassPlus"]}, "usage": {"quota":
[{"totalVolume": "100", "Type": "quota", "name": "Weekend", "nextResetTime": "2
```

```
014-01-10T02:00:00Z", "cid": "9223372036854775808", "QuotaState": "active"},
{"totalVolume": "100", "Type": "quota", "name": "Evenings", "nextResetTime": "2
014-02-01T00:00:00Z", "cid": "9223372036854775809", "QuotaState": "active"}]
, "version": 3}, "definition": {"DynamicQuota":
[{"InstanceId": "15678", "Priority": "4", "InitialInputVolume": "15000", "Inte
rimReportingInterval": "1000", "Duration": "100", "InitialOutputVolume": "500
0", "Type": "pass", "activationdatetime": "2015-03-09T11:20:32", "expirationd
atetime": "2015-04-09T11:20:32", "InitialTotalVolume": "20000", "name": "Dyna
micAggregateLimit", "InitialServiceSpecific": "9", "InitialTime": "135"}], "S
equenceNumber": 1, "version": 1}, "state": {"property":
[{"name": "mcc", "value": "123"},
{"name": "expire", "value": "201402-09T11:20:32"}], "version": 1}}}]
{"Timestamp": "Mon, 22 Jan 2024 11:00:54 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1705921254488-61110498-0699-420d-
baae-770fb5623a13", "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-
f6a55a328b03", "method": "PUT", "response": "400 BAD_REQUEST"}
```

- **PATCH:**

- Success request and response:

```
{"timeStamp": "Mon, 22 Jan 2024 10:54:24 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1705920864822-61110498-0699-420d-
baae-770fb5623a14", "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-
f6a55a328b03", "method": "PATCH", "request": "[{"op": "add", "patchFields":
{"Entitlement": ["DayPass"]}, "filters": null},
{"op": "replace", "patchFields": {"MSISDN":
["14161234567"]}, "filters": null}, {"op": "delete", "patchFields":
{"Entitlement": null, "filters": null}]}]
{"Timestamp": "Mon, 22 Jan 2024 10:54:24 UTC", "LogType": "OCUDR-PROV-
LOGS", "correlationId": "1705920864822-61110498-0699-420d-
baae-770fb5623a14", "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-
f6a55a328b03", "method": "PATCH", "response": "204 NO_CONTENT"}
```

- Failure request and response:

```
{"timeStamp": "Mon, 22 Jan 2024 11:02:07 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1705921327563-61110498-0699-420d-
baae-770fb5623a15", "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-
f6a55a328b03", "method": "PATCH", "request": "[{"op": "add", "patchFields":
{"Entitlement": ["DayPass"]}, "filters": null},
{"op": "replace", "patchFields": {"MSISDN":
["14161234567"]}, "filters": null}, {"op": "delete", "patchFields":
{"Entitlement": null, "filters": null}]}]
{"Timestamp": "Mon, 22 Jan 2024 11:02:07 UTC", "LogType": "OCUDR-PROV-
LOGS", "correlationId": "1705921327563-61110498-0699-420d-
baae-770fb5623a15", "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-
f6a55a328b03", "method": "PATCH", "response": "400 BAD_REQUEST"}
```

- **DELETE:**

- Success request and response:

```
{"timeStamp": "Mon, 22 Jan 2024 10:55:26 UTC", "logType": "OCUDR-PROV-
LOGS", "correlationId": "1705920926635-61110498-0699-420d-
baae-770fb5623a16", "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-
f6a55a328b03", "method": "DELETE", "request": ""}
```

```
{
  "Timestamp": "Mon, 22 Jan 2024 10:55:26 UTC",
  "logType": "OCUDR-PROV-LOGS",
  "correlationId": "1705920926635-61110498-0699-420d-baae-770fb5623a16",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "method": "DELETE",
  "response": "204 NO_CONTENT"
}
```

- Failure request and response:

```
{
  "timeStamp": "Mon, 22 Jan 2024 11:02:51 UTC",
  "logType": "OCUDR-PROV-LOGS",
  "correlationId": "1705921371341-61110498-0699-420d-baae-770fb5623a17",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "method": "DELETE",
  "request": ""
}
{
  "Timestamp": "Mon, 22 Jan 2024 11:02:51 UTC",
  "logType": "OCUDR-PROV-LOGS",
  "correlationId": "1705921371341-61110498-0699-420d-baae-770fb5623a18",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "method": "DELETE",
  "response": "404 NOT_FOUND"
}
```

Managing Provisioning Logs

Enable

You can enable or disable the provisioning logs and set the supported provisioning APIs using Helm configuration:

- Set the `provLogsEnabled` parameter to `true` or `false` in the `nudr-dr-provservice` section of the `ocudr-custom-values.yaml` file to enable or disable the provisioning logs. The default value is set as `false`. For more information about this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- Set the `provLogsApiNames` parameter to the supported provisioning APIs. The default value for `provLogsApiNames` parameter is `nudr-dr-prov`. The supported provisioning APIs are as follows:
 - `nudr-dr-prov`
 - `nudr-group-id-map-prov`
 - `slf-group-prov`
 - `n5g-eir-prov`

Note

When you enable the provisioning logs feature, the log storage value for `nudr-dr-prov` must be set to 4000 to accommodate provisioning logs.

```
# nudr-dr-provservice microservice configurations
nudr-dr-provservice:
# nameOverride: "nudr-dr-provservice"
# Image Details
enabled: *drProvisioningEnabled
onDemandMigrationProvEnabled : false
provLogsEnabled : false
provLogsApiNames : "nudr-dr-prov"
image:
  name: nudr_datarepository_service
  tag: ${nudr_drservice tag}
```

You can enable or disable the provisioning logs and set the supported provisioning APIs name using CNC Console:

- Set the `provLogsEnabled` parameter to true or false in the [Provisioning Data Repository Service](#) in CNC Console.
- Set the `provLogsApiNames` parameter to the supported provisioning APIs in the [Provisioning Data Repository Service](#) in CNC Console.

Configure using REST API:

You can perform the PUT operation on the `http://10.75.229.173:31252/nudr-config/v1/udr.nudrprovservice.cfg/DR-PROV-SERVICE` to enable or disable the provisioning logs feature and set the `provLogsApiNames` parameter to the supported provisioning APIs. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "provLogsEnabled": true,
  "hikariPoolSize": 25,
  "onDemandMigrationEnabled": false,
  "tracingEnabled": false,
  "provLogsApiNames": [
    "nudr-dr-prov"
  ]
}
```

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

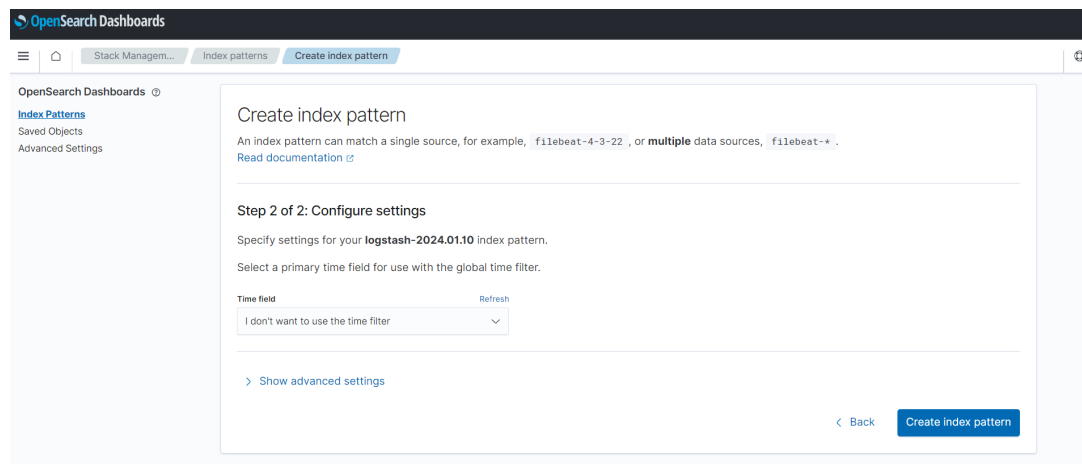
Accessing Provisioning Logging

You can access the provisioning logging by using OpenSearch or by copying logs from the running pod.

Accessing Provisioning Logging Using OpenSearch

OpenSearch is a community driven, open-source search used to ingest, view, and search data. Perform the following steps to access the provisioning logs.

1. On OpenSearch Dashboards GUI, click the **Hamburger** icon on the top left corner to open the sidebar menu.
2. Expand the **Management** section and select **Stack Management**.
3. Select **Index Patterns** and click **Create index pattern**.
4. Enter the name of index pattern in the **Index pattern name** field.
5. Verify that you get a "Your index pattern matches <n> sources" message and index patterns in the **logstash-YYYY.MM.DD** format as seen in the following image:



6. Click **Next step**.
7. Select **I don't want to use the time filter** and click **Create index pattern**.
8. Ensure that the web page containing the indices appears on the main viewer frame.
9. Click the **Hamburger** icon on the top left corner to open the sidebar menu and select **Discover**.
10. Select your index from the drop-down. Add the required filter and search for logs.
11. Select the log to remove all the remaining fields from filtered provisioning logs.
12. Click **Save** to save the final filter logs.
13. Click the **Reporting option** or **View Report** to download the logs.

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

Accessing Provisioning Logging by Copying the Logs from the Pod

Perform the following steps to copy the provisioning logs from the pod:

1. Run the following commands to get list of nudr-dr-provservice pods.

```
kubectl get pods -n <namespace> | grep 'nudur-dr-provservice'
```

The following pods will be listed.

```
ocudr-nudur-dr-provservice-d48f787df-l5nbd 1/1 Running 0 111m
```

2. Run the following command to log in to the pod. Go to the path `home/udruser/provlogs/` to get list of provisioning log files.

```
kubectl exec -it ocudr-nudur-dr-provservice-d48f787df-l5nbd -n <namespace>
bash
```

```
cd home/udruser/provlogs/
```

```
ls
```

```
provision.log provision.1.log.gz provision.2.log.gz
```


3. Run following command to copy the provisioning log files from the pod. You must perform this step for all the files listed in [Step 2](#).

```
kubectl cp ocudr-nudr-dr-provservice-d48f787df-l5nbd:home/udruser/provlogs/
provision.log provision.log -n prakash
```

```
kubectl cp ocudr-nudr-dr-provservice-d48f787df-l5nbd:home/udruser/provlogs/
provision.1.log.gz provision.1.log.gz -n prakash
```

```
kubectl cp ocudr-nudr-dr-provservice-d48f787df-l5nbd:home/udruser/provlogs/
provision.2.log.gz provision.2.log.gz -n prakash
```

4. You must repeat the [Step 2](#) and [Step 3](#) for all the pods listed in [Step 1](#).
5. Transfer all the log files to a remote path.

2.11 Error Response and Logging Enhancement

This section describes the Error Response and Logging Enhancement feature.

2.11.1 Error Response Enhancement

UDR sends error responses to consumer NFs due to some internal exceptions, such as signaling, request and url validation errors, connection failure, timeout, or unknown host while processing message requests from consumer NFs. These error responses have payloads containing the problem title, status, details, and cause of the error that are used to investigate the error. The error responses earlier used to contain only the error description in the details field, which was insufficient to troubleshoot the errors.

```
{
  "title": "Service Unavailable",
  "status": "503",
  "details": "Currently Congested. Could not process request.",
  "cause": "NF_CONGESTION"
}
```

With the enhanced error response mechanism, UDR sends additional pieces of information such as server FQDN, NF service name, vendor name, and error ID in the details field of the payload to investigate the errors.

The new format of the payload is as follows:

```
{
  "title": "[errorcodeProfiles[i].errorTitle]",
  "status": "[errorcodeProfiles[i].status]",
  "details": "[Server-FQDN]: [NF Service]:
[errorcodeProfiles[i].errorDescription]: [VendorNF**-App-Error ID]",
  "cause": "[errorcodeProfiles[i].errorCause]"
}
```

Example:

```
{
  "title": "Not Found",
```

```

    "status": 404,
    "detail": "slf01.abc.com: Nudr_GroupIDmap: Group Not Found: OSLF-DRS-SIG-
E302",
    "cause": "GROUP_IDENTIFIER_NOT_FOUND",
    "invalidParams": null
  }

```

Sample ProblemDetails.detail as follows:

<nfFqdn>: <service-name>: <Readable Error details>: <App-Error-Code>, example,
slf01.abc.com: Nudr_GroupIDmap: Request header Issue, Unsupported Media Type:
OSLF-DRS-HDRVLD-E001

The following table describes the parameters of the details field of the payload:

Table 2-11 Parameters of the Details Field of the Payload

Parameter Name	Description
nfFqdn	Indicates the NF FQDN. It is obtained from the nfFqdn Helm Chart parameter. Sample Value: slf01.abc.com: Nudr_GroupIDmap
service-name	Indicates the microservice name. It is the originator of the error response. This value is static and cannot be configured. Sample Value: Nudr_GroupIDmap
Readable Error details	Provides a short description of the error. Sample Value: Group Not Found
App-Error ID	Indicates the microservice ID and the error IDo<nf type>-<serviceId>-<category>-E<XXX>. Sample Value: OSLF-DRS-SIG-E302, where, • OSLF is the vendor NF • DRS is the microservice ID • SIG is the category • E302 is the app error code
nftype	Indicates the vendor or NF type. This parameter is prefixed with "O", which indicates Oracle. For example, if the NF type is SLF, the vendor name becomes OSLF. It is obtained from the nfType Helm Chart parameter. Sample Value: OSLF
serviceId	It is either DRS (nudr-drservice) or DRP (nudr-dr-provservice). This value is set based on container name.
Category	Category to be used is fetched from error catalog. Errors are classified into categories based on serviceid. Following are the list of categories: • SIG • PROV • URIVLD • HDRVLD • REQVLD • DB • INTRNL

For the list of Error Code details, possible causes, and corrective actions, see [Error Code Catalogue](#).

Enable

This feature is enabled automatically at the time of deployment.

Configure

The following configurations are introduced to include the `nffqdn` and `nfType` on the `problemDetails.detail` parameter:

Table 2-12 Helm Configurations

Configurations	Description
nudr-drservice.nfType nudr-dr-provservice.nfType ingressgateway-prov.global.nfType ingressgateway-sig.global.nfType	It is the configuration used to configure the NF type using the Helm charts for error detail. It is part of global section of the <code>ocudr-custom-values.yaml</code> file.
nudr-drservice.nfFqdn nudr-dr-provservice..nfFqdn ingressgateway-prov.global.nfFqdn ingressgateway-sig.global.nfFqdn	It is the configuration used to configure the NF FQDN using Helm charts for error detail. It is part of global section of the <code>ocudr-custom-values.yaml</code> file.

Observe

There are no metrics, alerts, or KPIs for this feature.

2.11.2 Error Logging Enhancement

With the Error Response Logging feature, additional attributes are added to the existing ERROR logs which gets populated with appropriate values during failure scenarios. The new attributes that are added as part of this feature are `errorStatus`, `errorTitle`, `errorDetails`, `errorCause`, `sender`, `receiver`, and `subscriberId`. Among these `errorStatus`, `errorTitle`, `errorDetails`, and `errorCause` attributes are fetched from the existing `ProblemDetails` attribute. The feature is enabled using the `additionalErrorLogging` and `logSubscriberInfo` feature flags in the logging API for each microservice. By default, the new attributes are added to the logs. These new attributes is populated with appropriate values in failure scenarios only when the feature is enabled.

Sample REST API body as follows:

```
{
  "appLogLevel": "WARN",
  "packageLogLevel": [
    {
      "packageName": "jetty",
      "logLevelForPackage": "WARN"
    },
    {
      "packageName": "springframework",
      "logLevelForPackage": "WARN"
    },
    {
      "packageName": "root",
```

```

        "logLevelForPackage": "WARN"
    },
    ],
    "logSubscriberInfo": "DISABLED",
    "additionalErrorLogging": "DISABLED"
}

```

Sample logs for microservices are as follows:

- **nudr-datarepository-service:**

Sample_logs_dr_signaling

```

{"instant":
{"epochSecond":1718770954,"nanoOfSecond":523726788},"thread":"XNIO-1
task-2","level":"ERROR","loggerName":"ocudr.udr.services.exception.Customiz
edResponseEntityExceptionHandler","message":"Error Response

Details","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.Abstr
actLogger","threadId":302,"threadPriority":5,"errorStatus":"404","errorTit
le":"Not
Found","errorDetails":"ocudr-ingressgateway-sig.oracle.com:
Nudr_DataRepository: Subscriber does not exist:
OUDR-DRS-SIG-
E272","errorCode":"USER_NOT_FOUND","sender":"UDR-5a7bd676-ceeb-44bb-95e0-
f6a55a328b03
ocudr-ingressgateway-sig.oracle.com","receiver":"OCATS-
UDR","messageTimestamp":"2024-06-19T04:22:34.523+0000","application":"ocudr
","engVersion":"24.2.0-
rc.50","mktgVersion":"24.2.0.0.0","microservice":"ocudr-nudr-
drservice","vendor":"Oracle","resourceId":"nudr-dr/v2/policy-data/ues/sm-
data","resultCode":"404","ocLogId":"1718770952081_520_ocudr-ingressgateway-
sig-57d49b5767-fk4ht","requestType":"GET"}

```

Sample_logs_dr_provisioning

```

{"instant":
{"epochSecond":1718771274,"nanoOfSecond":165140206},"thread":"XNIO-1
task-2","level":"ERROR","loggerName":"ocudr.udr.services.exception.Customiz
edResponseEntityExceptionHandler","message":"Error Response

Details","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.Abstr
actLogger","threadId":291,"threadPriority":5,"errorStatus":"400","errorTit
le":"Bad
Request","errorDetails":"ocudr-ingressgateway-sig.oracle.com:
Nudr_DataRepository: Payload should not be empty:
OUDR-DRP-REQVLD-
E102","errorCode":"MANDATORY_IE_INCORRECT","sender":"UDR-5a7bd676-
ceeb-44bb-95e0-f6a55a328b03
ocudr-ingressgateway-sig.oracle.com","receiver":"OCATS-
UDR","messageTimestamp":"2024-06-19T04:27:54.165+0000","application":"ocudr
","engVersion":"24.2.0-
rc.50","mktgVersion":"24.2.0.0.0","microservice":"ocudr-nudr-dr-
provservice","vendor":"Oracle","subscriberId":"msisdn-4617722711377","resou
rceId":"nudr-dr-prov/v2/policy-

```

```
data", "resultCode": "400", "ocLogId": "1718771274068_466_ocudr-ingressgateway-
prov-7bf8c864cd-5dv47", "requestType": "PUT"}
```

- **nudr-ondemand-migration:**

```
{"instant":
{"epochSecond":1718771522,"nanoOfSecond":969319819},"thread":"XNIO-1
task-2","level":"ERROR","loggerName":"ocudr.migration.ondemand.exception.Cu
stomizedResponseEntityExceptionHandler","message":"Error Response

Details","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLog
ger","threadId":434,"threadPriority":5,"errorStatus":"404","errorTitle":"No
t
Found","errorDetails":"ocudr-ingressgateway-sig.oracle.com: nudr-
ondemand-migration: Subscriber does not
exist: OUDR-ODM-INTRNL-
E272","errorCause":"USER_NOT_FOUND","sender":"UDR-5a7bd676-ceeb-44bb-95e0-
f6a55a328b03
ocudr-ingressgateway-sig.oracle.com","receiver":"UDR-5a7bd676-
ceeb-44bb-95e0-f6a55a328b03
ocudr-ingressgateway-
sig.oracle.com","messageTimestamp":"2024-06-19T04:32:02.969+0000","applicat
ion":"ocudr","engVersion":"24.2.0-
rc.50","mktgVersion":"24.2.0.0.0","microservice":"ocudr-nudr-ondemand-
migration","vendor":"Oracle","subscriberId":"msisdn-59376755","ocLogId":"17
18771522954_434_ocudr-nudr-ondemand-migration","requestType":"GET"}
```

- **nudr-config:**

```
{"instant":
{"epochSecond":1718771822,"nanoOfSecond":458471958},"thread":"XNIO-1task-6"
,"level":"ERROR","loggerName":"com.oracle.nf.common.configurationService.ex
ception.CustomizedResponseEntityExceptionHandler","message":"Error Response

Details","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLog
ger","threadId":190,"threadPriority":5,"errorStatus":"403","errorTitle":"Fo
rbidden","errorDetails":"nudr-config: Read only field can not be modified,
MODIFICATION_NOT_ALLOWED: OUDR-DRC-CFG-
E559","errorCause":"MODIFICATION_NOT_ALLOWED","sender":"UDR-5a7bd676-
ceeb-44bb-95e0-f6a55a328b03","receiver":"PostmanRuntime/
7.37.3","messageTimestamp":"2024-06-19T04:37:02.458+0000","application":"oc
udr","engVersion":"24.2.0-
rc.50","mktgVersion":"24.2.0.0.0","microservice":"ocudr-nudr-
config","vendor":"Oracle","resourceId":"/nudr-config/v1/
udr.nudrservice.cfg/DR-
SERVICE","resultCode":"400","ocLogId":"1718771822446_190_ocudr-nudr-
config-549cdd85f9-vjf8r","requestType":"PUT"}
```

- **nudr-bulk-import:**

```
{"instant":
{"epochSecond":1714996478,"nanoOfSecond":833348071},"thread":"Async-Writer-
Thread -
7","level":"ERROR","loggerName":"ocudr.udr.bulkimport.util.ImportToolHelper
","message":"Error Response

Details","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.Abst
```

```
ractLogger", "threadId":129, "threadPriority":5, "errorStatus":"400", "errorTit
le":"Bad Request", "errorDetails":"udr001.oracle.com: Nudr_DataRepository:
Invalid URI sent from client: OUDR-DRP-URIVLD-
E021", "errorCause":"MANDATORY_IE_INCORRECT", "messageTimestamp":"2024-05-06T
11:54:38.833+0000", "application":"blk", "engVersion":"24.2.0-
beta.46", "mktgVersion":"24.1.0.0.0", "microservice":"blk-nudr-bulk-
import", "vendor":"Oracle", "resourceId":"http://ocudr-ingressgateway-
prov.prak:80/n5g-eir-prov/v1/equipment-status/
imei-399236049675884", "resultCode":"400", "ocLogId":"1714996478722_328_blk-
nudr-bulk-import-0", "requestType":"PUT"}
```

- **nudr-notify-service:**

```
{"instant":
{"epochSecond":1720098082, "nanoOfSecond":762665335}, "thread": "@575e572f-61"
, "level": "ERROR", "loggerName": "ocudr.udr.services.exception.CustomizedRespo
nseEntityExceptionHandler", "message": "Error Response
Details", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.Abstr
actLogger", "threadId": 61, "threadPriority": 5, "errorStatus": "503", "errorTitl
e": "Service Unavailable", "errorDetails": "ocudr-ingressgateway-
sig.oracle.com: egressgateway: Service Unavailable: OUDR-EGW-
E003", "errorCause": "Encountered unknown host exception at
EGW", "sender": "UDR-5a7bd676-ceed-44bb-95e0-f6a55a328b03 ocudr-
ingressgateway-sig.oracle.com", "receiver": "UDR-5a7bd676-ceed-44bb-95e0-
f6a55a328b03 ocudr-ingressgateway-
sig.oracle.com", "messageTimestamp": "2024-07-04T13:01:22.762+0000", "applicat
ion": "ocudr-shriats", "engVersion": "24.2.0-
rc.50", "mktgVersion": "24.2.0.0.0", "microservice": "ocudr-shriats-nudr-
notify-service", "vendor": "Oracle"}
```

Managing the Error Logging Enhancements

Enable

You can enable the Error Logging Enhancements using Helm Configuration

You can enable or disable the feature by setting the `additionalErrorLogging` parameter to enable or disable in the logging section for each microservices. The default value is set as disabled.

You can enable or disable the logging of UE Identifier and subscriberid by setting the `logSubscriberInfo` parameter to enable or disable in the logging section for each microservices. The default value is set as disabled.

```
# Logging level
logging:
additionalErrorlogging: "DISABLED"
logSubscriberInfo: "DISABLED"
level:
root: "WARN"
```

Configure

You can configure this feature using REST API and the CNC Console.

- **Configure using CNC Console:** Enable the Additional Error Logging and Log Subscriber Info parameter as enabled for each of the microservice in the [Logging Level Configuration](#) in the CNC Console.
- **Configure using REST API:** Enable the `additionalErrorLogging` and `logSubscriberInfo` parameter as `ENABLED` for each of the microservices. The default value is set as `DISABLED`. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "appLogLevel": "WARN",
  "packageLogLevel": [
    {
      "packageName": "jetty",
      "logLevelForPackage": "WARN"
    },
    {
      "packageName": "springframework",
      "logLevelForPackage": "WARN"
    },
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    }
  ],
  "logSubscriberInfo": "DISABLED",
  "additionalErrorLogging": "DISABLED"
}
```

Upgrade Impacts

- When you upgrade UDR from previous release version by setting `additionalErrorLogging` parameter as `ENABLED`, then additional information such as ProblemDetails IE is provided in all the published logs with error response details.
- When you upgrade UDR from previous release version by setting `logSubscriberInfo` parameter as `ENABLED`, then `subscriberId` is provided in all the published logs.

Rollback Impacts

When you rollback UDR from previous release version the configuration is also rolled back and Error Logging Enhancement feature does not work.

Log Throttling using Burst Filter

Burst filter is used to throttle the log when the log publishing rate exceeds the configured rate. The configured values are mentioned over a period of time and for maximum it is per second, and any excess logs after the rate is reached is ignored.

- **level:** It is the log level below which logs will be throttled. For example, `DEBUG` level will throttle `TRACE` and `DEBUG` logs.
- **rate:** It is the average log printing rate over a period of time after which the log throttling starts.
- **max:** It is the maximum number of logs allowed per second in case of maximum burst scenarios.

The default configuration is as follows:

logging:

```
burstFilter:
  level: "DEBUG"
  rate: 500
  max: 3000
```

Note

The time window is calculated as max/rate . For example, $3000/500$ is six seconds. For six seconds you will not receive over 500 logs.

The level configuration of the burst filter is not similar to the default logging behavior in UDR. Example of level configuration is as follows:

- **ERROR:** ERROR, WARN, INFO, and DEBUG are filtered
- **WARN:** WARN, INFO, and DEBUG are filtered
- **INFO:** INFO and DEBUG are filtered
- **DEBUG:** DEBUG is filtered

Observe

There are no metrics, alerts, or KPIs for this feature.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.12 Ingress Gateway Pod Protection

This feature protects the Ingress Gateway pods from congestion conditions due to uneven traffic distribution, network fluctuations, traffic bursts, or unexpected high traffic volume. It ensures the protection and mitigation of Ingress Gateway pods from entering congestion condition and also facilitating necessary actions for recovery. As a front end microservice for HTTP traffic, it is important for Ingress Gateway to have pod protection implemented.

The pod protection is performed based on the CPU consumption of the Ingress Gateway Pods as explained in the Congestion State Parameters. It is measured at different load states mentioned in the Ingress Gateway Load States.

Note

Ingress Gateway Pod Protection supports only UDR and EIR mode. For SLF mode support refer to [Ingress Gateway Pod Protection Using Rate Limiting](#).

Congestion State Parameters**CPU Usage**

In the Pod Protection feature, each Ingress Gateway microservice pod monitors its congestion state. This state is tracked in terms of CPU consumption, measured in nanoseconds, using Kubernetes cgroup (cpuacct.usage).

It is periodically monitored and calculated using the following formula. Then, it is compared against the CPU thresholds configured through the Rest API to determine the congestion state. For more information about the parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

$$\frac{\text{CurrentCpuUsage} - \text{LastCpuUsage}}{\text{CurrentTime} - \text{LastSampleTime}} \times 100$$

$$\text{CPUs}$$

Where,

CurrentCpuUsage is the counter reading at current periodic cycle.

LastCpuUsage is the counter reading at previous periodic cycle.

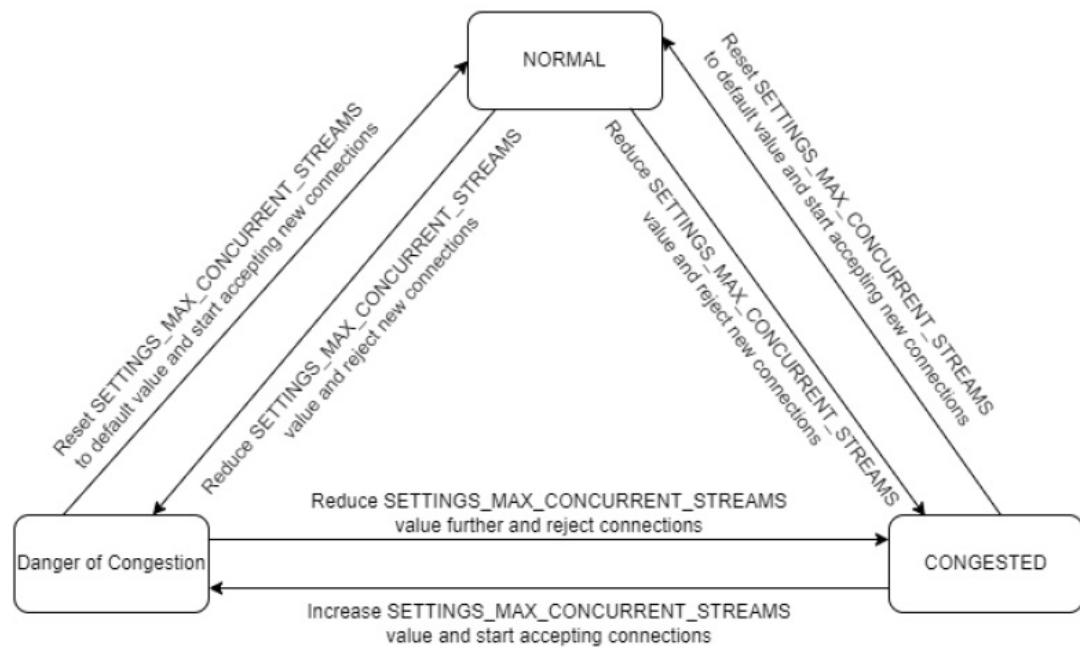
CurrentTime is the current time snapshot.

LastSampletime is the previous periodic cycle time snapshot.

CPUs is the total number of CPUs for a given pod.

Ingress Gateway Load States

The following states are used to detect overload conditions. This ensures the protection and mitigation of pods from entering an overload condition, while also facilitating necessary actions for recovery.



Note

The transition can occur between any states based on the congestion parameters. The threshold for these congestion parameters is preconfigured and must not be changed.

- Congested State:** This is the upper bound state where the pod is congested. This means one or more congestion parameters are above the configured thresholds for the congested state. For more information about the configuration using REST API, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*. The pod can be transitioned to the Congested State either from the Normal State or the DoC state. When the pod reaches this state, the following actions are performed:
 - new incoming HTTP2 connection requests are not accepted.
 - the pod gradually decrements the number of concurrent streams by updating `SETTINGS_MAX_CONCURRENT_STREAMS` parameter in a `SETTINGS` frame to the configured `maxConcurrentStreamsPerCon` value at a regular interval. The concurrent streams are decremented based on the value configured in `decrementBy` parameter. And, the regular interval is configured in the `decrementSamplingPeriod` parameter.
- Danger of Congestion (DOC):** This is the intermediate state where the pod is approaching a congested state. This means if CPU is above the configured thresholds for the DoC state.
 - any new incoming HTTP2 connection requests are not accepted.
 - if the pod is transitioning from the Normal State to the DoC state, the pod gradually decrements the number of concurrent streams by updating `SETTINGS_MAX_CONCURRENT_STREAMS` parameter in a `SETTINGS` frame to the configured `maxConcurrentStreamsPerCon` value at a regular interval. The concurrent streams are decremented based on the value configured in `decrementBy` parameter. And, the regular interval is configured in the `decrementSamplingPeriod` parameter.

- if the pod is transitioning from the Congested State to the DoC state, the pod gradually increments the number of concurrent streams by updating `SETTINGS_MAX_CONCURRENT_STREAMS` parameter in a `SETTINGS` frame to the configured `maxConcurrentStreamsPerCon` value at a regular interval. The concurrent streams are incremented based on the value configured in `incrementBy` parameter. And, the regular interval is configured in the `incrementSamplingPeriod` parameter.

Note

In any service mesh based deployment, if any incoming connections to the pod get terminated at the sidecar container then the sidecar container creates a new connection toward the application container. These incoming connections from the peer are managed by the sidecar and outside the purview of the application container.

Hence when the Ingress Gateway container reaches DOC or Congested level, in a service mesh based deployment, the Ingress Gateway container will only be able to stop accepting new connections from the sidecar container. Also in this state, the Ingress Gateway container will reduce the concurrency of the existing connections between the sidecar container and the Ingress Gateway container.

- **Normal State:** This is the lower bound state where all the congestion parameters for the pod are below the configured thresholds for DoC and Congested states. When the pod reaches this state, the following actions are performed:
 - the pod will continue accepting new incoming HTTP2 connection requests.
 - the pod will continue accepting requests on the existing HTTP2 connections.
 - in case the pod is transitioning from the Congested or DoC state to Normal state, the pod gradually increments the number of concurrent streams by updating `SETTINGS_MAX_CONCURRENT_STREAMS` parameter in a `SETTINGS` frame to the configured `maxConcurrentStreamsPerCon` value at a regular interval. The concurrent streams are incremented based on the value configured in `incrementBy` parameter. And, the regular interval is configured in the `incrementSamplingPeriod` parameter.

To avoid toggling between these states due to traffic pattern, it is required for the pod to be in a particular state for a given period before transitioning to another state. The below configurations are used to define the period till which the pod has to be in a particular state:

- `stateChangeSampleCount`
- `monitoringInterval`

Formula for calculating the period is as follows:

$(stateChangeSampleCount * monitoringInterval)$

For more information about the configuration using REST API, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Pending Requests

Pending request count is the number of pending requests that are waiting for a response. The counter is incremented for all incoming requests and decremented when there is a response.

The following table provides the threshold configurations.

Table 2-13 Threshold configurations

Type	CPU and Pending Count	Description
Normal	Threshold start CPU: 0 Threshold end CPU: 79% Threshold start Pending Count : 0 Threshold end Pending Count : 4999	New connections will be accepted in normal state
DOC Alert	Threshold start CPU: 80% Threshold end CPU: 89% Threshold start Pending Count : 5000 Threshold end Pending Count : 6999	New connection will be rejected in DOC STATE.
Congested Alert	Threshold start CPU: 90% Threshold end CPU: 100% Threshold start Pending Count : 7000 Threshold end Pending Count : 7000+	New connection will be rejected in CONGESTED STATE.

Managing Ingress Gateway Pod Protection

This section explains the procedure to enable and configure the feature.

Enable

You can enable this feature using REST API. Perform the REST API configurations as explained below:

1. Use the API path as `http://<nudr-config svc IP>:<port>/udr/nf-common-component/v1/igw-prov/podprotection` or `http://<nudr-config svc IP>:<port>/udr/nf-common-component/v1/igw-sig/podprotection`.
2. Set `enabled` and `congestionControl.enabled` to `true`.

3. Note

You should not use memory as a congestion state parameter for pod protection feature because the memory accumulated by the pod during the traffic run is not released by the pod even when the traffic is reduced. This causes the pod to remain in DOC or CONEGSTED state.

Run the API using PUT method with the proposed values given in the Rest API. For more information about API path, see "Configuration APIs for Common Services" section in *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Given below is a sample REST API configuration to enable this feature:

```
{
  "enabled": true,
  "congestionControl": {
```

```

"states": [
  {
    "name": "Normal",
    "weight": 0,
    "entryAction": [
      {
        "action": "MaxConcurrentStreamsUpdate",
        "arguments": {
          "incrementBy": 30,
          "maxConcurrentStreamsPerCon": 100,
          "incrementByActionSamplingPeriod": 3
        }
      },
      {
        "action": "AcceptIncomingConnections",
        "arguments": {
          "accept": true
        }
      }
    ]
  },
  {
    "name": "DoC",
    "weight": 1,
    "entryAction": [
      {
        "action": "AcceptIncomingConnections",
        "arguments": {
          "accept": false
        }
      },
      {
        "action": "MaxConcurrentStreamsUpdate",
        "arguments": {
          "decrementBy": 30,
          "incrementBy": 30,
          "maxConcurrentStreamsPerCon": 50,
          "decrementByActionSamplingPeriod": 1,
          "incrementByActionSamplingPeriod": 3
        }
      }
    ],
    "resourceThreshold": {
      "cpu": 60,
      "pendingMessage": 5000
    }
  },
  {
    "name": "Congested",
    "weight": 2,
    "entryAction": [
      {
        "action": "AcceptIncomingConnections",
        "arguments": {
          "accept": false
        }
      }
    ]
  }
]

```

```

        },
        {
            "action": "MaxConcurrentStreamsUpdate",
            "arguments": {
                "decrementBy": 30,
                "maxConcurrentStreamsPerCon": 10,
                "decrementByActionSamplingPeriod": 1
            }
        }
    ],
    "resourceThreshold": {
        "cpu": 75,
        "pendingMessage": 7000
    }
}
},
"enabled": true,
"actionSamplingPeriod": 3,
"stateChangeSampleCount": 10
},
"monitoringInterval": 100
}

```

Observe

Metrics

The following metrics are used to provide information about this feature:

- `oc_ingressgateway_pod_congestion_state{namespace='$namespace'}`
- `oc_ingressgateway_pod_resource_state{type="CPU/PENDING_REQUEST",namespace='$namespace'}`
- `oc_ingressgateway_pod_resource_stress{type="CPU/PENDING_REQUEST",namespace='$namespace'}`
- `oc_ingressgateway_incoming_pod_connections_rejected_total{namespace="$namespace"}`
- `oc_ingressgateway_pod_congestion_state{namespace='$namespace'}`
- `oc_ingressgateway_incoming_pod_connections_rejected_total{namespace="$namespace"}`

For information about the metrics, see [Ingress Gateway Metrics](#).

Alerts:

The following alerts generated for this feature:

- [IngressgatewayPodProtectionDocState](#)
- [IngressgatewayPodProtectionCongestedState](#)

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.13 Support for cnDBTier APIs in CNC Console

With the implementation of this feature, cnDBTier APIs are integrated into the CNC Console, and UDR users can view specific cnDBTier APIs, such as checking the available backup lists, cnDBTier version, database statistics, heartbeat status, local cnDBTier cluster status, georeplication status, and initiating on-demand backup on the CNC Console.

The following cnDBTier APIs can be viewed in CNC Console:

- Backup List: This API displays the details of stored backups, such as the ID and size of the backup.
- cnDBTier Backup Status: This API displays the cnDBTier backup status, such as the current timestamp, backup in progress, and next scheduled backup.
- cnDBTier version: This API displays the cnDBTier version.
- Database Statistics Report: This API displays the number of available database.
- Georeplication Recovery: This API provides the recovery status of georeplication for the cnDBTier cluster:
 - Update Cluster As Failed: This API is used to mark a disrupted cluster as failed.
 - Start Georeplication Recovery: This API is used to start the georeplication recovery process for any failed cluster. Only one cluster can be recovered at a time.
 - Georeplication Recovery Status: This API is used to monitor the recovery status of georeplication for both FAILED and ACTIVE cnDBTier sites.
- Geo Replication Status:
 - Real Time Overall Replication Status: This API displays the overall replication status in multisite deployments. For example, in a four-site deployment, it provides the replication status between the following sites: site1-site2, site1-site3, site1-site4, site2-site3, site2-site4, and site2-site1. This is applicable for all other sites.
 - Site Specific Real Time Replication Status: This API displays the site-specific replication status.
- HeartBeat Status: This API displays the connectivity status between the local site and the remote site name to which UDR is connected.
- Local Cluster Status: This API displays the status of the local cluster.
- On-Demand Backup: This API displays the status of initiated on-demand backups.
- cnDBTier Health: This API displays the health status of the available databases.
 - Backup Manager Health Status: This API displays the health status of the backup manager service. It checks whether the backup manager service is up or not and if the service can connect to database or not.
 - Monitor Health Status: This API displays the health status of the monitor service. It checks the following:
 - * If the monitor service is up or not
 - * If the service can connect to database or not

- * If the metrics are fetched or not (the metrics are fetched when the service is up and vice versa)
- NDB Health Status: This API displays the health status of the replication service whether the pod is connected to PVC or not and checks whether the pods status is up or not.

Note

PVC is not applicable to some of the pods.

- Replication Health Status: This API displays the health status of the replication service. It checks whether the replication service is up or not and the service can connect to database or not.

**Managing cnDBTier APIs in CNC Console
Enable**

The CNC console section for accessing the cnDBTier APIs is available in UDR if the cnDBTier is configured as an instance during the CNC Console deployment. For more information about integrating cnDBTier APIs in CNC Console, see the "NF Single Cluster Configuration With cnDBTier Menu Enabled" section in *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

There is no option to disable this feature.

Configure

cnDBTier APIs can be accessed or configured using the CNC Console. For more information, see [cnDBTier](#) in the CNC Console.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

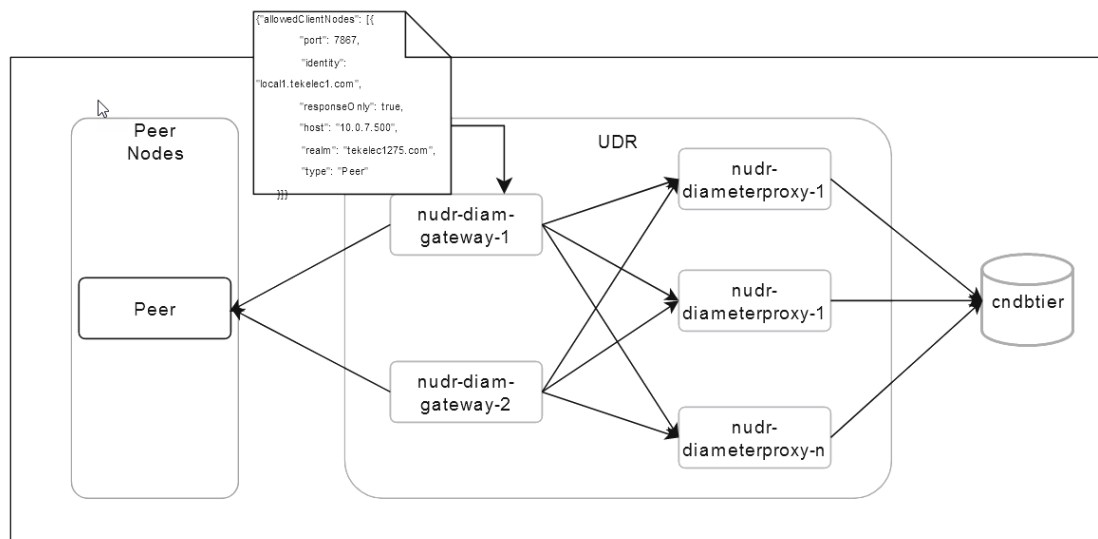
In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.14 Diameter Connection Configuration

The Diameter Connection Configuration feature provides the capability to trigger a Connection Establishment Request (CER) from the Diameter Gateway pod to each of the configured diameter endpoints. This feature enables the distribution of the diameter connection in a controlled manner by balancing the diameter traffic evenly between the Diameter Gateway pods. This is achieved by adding peer node details as a configuration in the Diameter Gateway.

Figure 2-6 Diameter Connection Configuration



When you configure the peer node details in the allowed client node list of the Diameter Gateway, the Diameter Gateway initiates one connection by sending Connection Establishment Request (CER) to the peer node from each Diameter Gateway pod. Diameter endpoints, such as Diameter Proxy and On-Demand Migration are added as peer nodes during deployment from the configuration map. Migration Tool configurations are added during tool installation using hooks.

Till UDR Release 23.2.0, the Diameter Gateway accepts connections only from the peer node. Since this is a Transmission Control Protocol (TCP) connection, there is only one connection from peer node to one of the Diameter Gateway pods. Due to this, traffic is not load balanced between the Diameter Gateway pods. In this feature, peer node details are configured in the Diameter Gateway. The Diameter Gateway uses these details to establish one connection from each of the Diameter Gateway pods to the peer node. This enables even distribution of traffic between the Diameter Gateway pods.

Configure using REST API:

In the Diameter Gateway configurations, `peerNodesForNotifications` and `migrationPeers` fields, which was used to store peer nodes details are removed. The `allowedClientNodes` field is used to store peer nodes details, peer node identity, and realm to allow incoming connection from peer nodes.

You can perform the PUT operation on the `http://10.75.229.85:31430/nuds-config/v1/uds.diamgateway.cfg/DIAMETER-GATEWAY` to update the peer node details.

If the `responseOnly` parameter is set to true, then Diameter Gateway initiates a connection to the peer node. If the parameter is set to false, the Diameter Gateway uses the peer node details for verifying and accepting the incoming connections. Parameters `host` and `port` are mandatory only if the `responseOnly` parameter is set to true. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```

{
  "realm": "oracle.com",
  "identity": "nuds.oracle.com",
  "reconnectDelay": 3,

```

```
"responseTimeout": 4,  
"connectionTimeOut": 3,  
"watchdogInterval": 6,  
"reconnectLimit": 50,  
"onDemandMigrationEnabled": false,  
"allowedClientNodes": [{  
  "port": 7867,  
  "identity": "local1.tekelec1.com",  
  "responseOnly": true,  
  "host": "10.0.7.500",  
  "realm": "tekelec1275.com",  
  "type": "4GUDROndemandConnection"  
}, {  
  "port": 3868,  
  "identity": "sh-server",  
  "host": "10.75.185.161",  
  "responseOnly": true,  
  "realm": "seagull",  
  "type": "server"  
}]  
}
```

2.15 Conflict Resolution

In a multisite deployment, during replication down time, UDR updates subscriber data on all the sites. When the replication is restored, the updates are resolved automatically to all the database tables by MySQL NDB Cluster in the cnDBTier. In some cases, automatic resolution by MySQL causes subscriber data conflicts and anomalies in the database model. This feature provides an application-defined resolution to resolve the subscriber data conflicts by adding a timestamp column to determine whether an update is required on the replica.

Conflict resolution must be enabled on all UDR, SLF, and EIR deployments. Conflict resolution includes multiple database schema updates and must be run with caution. The conflict resolution feature and database schema changes are not automatically enabled when upgraded to 23.3.0. This feature must be enabled manually.

① Note

- Conflict resolution feature must be enabled during maintenance activity to avoid service disruptions.
- Rollback of conflict resolution feature must be avoided.

To Enable

Perform the following steps to enable the conflict resolution for a clean installation:

Note

Following SQL files are available in `ocudr_custom_configtemplates` file:

- `ALL_mode_ndb_replication_insert.sql`
- `SLF_mode_ndb_replication_insert.sql`.
- `EIR_mode_ndb_replication_insert.sql`.
- `ALL_mode_ndb_replication_insert_UPGRADE.sql`.file
- `SLF_mode_ndb_replication_insert_UPGRADE.sql`
- `EIR_mode_ndb_replication_insert_UPGRADE.sql`.

For more information on how to download the package, see *Downloading Unified Data Repository Package* section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

1. Post installation of `cnDBTier` and before installing UDR, SLF, or EIR across the sites, do one of the following based on deployment mode, on one of the `ndbappmysqld` nodes on one of the sites. In the SQL files, replace the string `<database_name>` with the UDR subscriber database name, which is common to all UDR sites:

Note

In case of a two or three site setup, perform this step only on site one, on one of the `ndbappmysqld` nodes, preferably `ndbappmysql-0` node.

- Run the SQL commands in the `ALL_mode_ndb_replication_insert.sql` file, if installed in ALL mode.
 - Run the SQL commands in the `SLF_mode_ndb_replication_insert.sql` file, if installed in SLF mode.
 - Run the SQL commands in the `EIR_mode_ndb_replication_insert.sql` file, if installed in EIR mode.
2. Set the following configurations under `appinfo` section on all UDR sites to enable the database and replication monitoring from UDR deployment. This is also used by `dbcr` service before triggering an audit operation.

```
appinfo:
  watchMySQL:
    true

    replicationStatusCheck:
    true

    scrapeInterval:
    5

    dbStatusUri: http://mysql-cluster-db-monitor-svc.cndb-site1:8080/db-
tier/status/
    local
    realtimeDbStatusUri: http://mysql-cluster-db-monitor-svc.cndb-
```

```

site1:8080/db-tier/status/cluster/local/
realtime
  replicationUri: http://mysql-cluster-db-monitor-svc.cndb-site1:8080/db-
tier/status/
replication

  replicationInterval: 30

```

3. Set the IP/FQDN list of the mate sites ingressgateway-prov services in the following configuration under `nudr_dbcr_auditor_service` section. Enable the `dbConflictResolutionEnabled` parameter in the global section of the `ocudr-custom-values.yaml` file. For more information about the configuration, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation Guide*.

```

global:
  dbConflictResolutionEnabled: "true"

nudr_dbcr_auditor_service:
  # Provide MateSite IGW IP List, Comma separated values. Provide fqdn or
  IP with port
  mateSitesIgwIPList: 'http://ocudr-ingressgateway-prov.myudr2:80,http://
ocudr-ingressgateway-prov.myudr3:80'

```

4. Perform a helm upgrade of UDR using the helm upgrade command.

Note

Steps 2 to 4 must be performed on all UDR sites during deployment.

Perform the following steps to enable the feature post upgrade or installation:

1. To modifying the existing UDR database tables from `cnDBTier`, you must enable the `ndb_allow_copying_alter_table` parameter in the `cnDBTier` custom yaml file and perform a Helm upgrade.

Note

Before executing the scripts you must make sure that the subscriber size in the tables must have lesser entries than the `MaxDMLOperationsPerTransaction` parameter on MySQL. It is recommended to perform the following procedure before provisioning. For the existing subscribers you must configure the `MaxDMLOperationsPerTransaction` parameter accordingly before the following procedure is performed.

2. Once the `cnDBTier` Helm upgrade is successful, manually run the SQL commands mentioned on one of the following SQL files based on deployment mode, on one of the `ndbappmysql` nodes on one of the sites:

Note

In case of a two or three site setup, perform this step only on site one, on one of the `ndbappmysql` nodes, preferably `ndbappmysql-0` node.

- Run the SQL commands in the `ALL_mode_ndb_replication_insert_UPGRADE.sql`, if installed in all mode.
 - Run SQL commands in the `SLF_mode_ndb_replication_insert_UPGRADE.sql` file, if installed in SLF mode.
 - Run the SQL commands in the `EIR_mode_ndb_replication_insert_UPGRADE.sql` file, if installed in EIR mode.
3. Set the following configurations under `appinfo` section on all UDR sites to enable the database and replication monitoring from UDR deployment. This is also used by `dbcr` service before triggering an audit operation.

```

appinfo:
  watchMySQL:
    true

    replicationStatusCheck:
    true

    scrapeInterval:
    5

    dbStatusUri: http://mysql-cluster-db-monitor-svc.cndb-site1:8080/db-
tier/status/
local
    realtimeDbStatusUri: http://mysql-cluster-db-monitor-svc.cndb-
site1:8080/db-tier/status/cluster/local/
realtime
    replicationUri: http://mysql-cluster-db-monitor-svc.cndb-site1:8080/db-
tier/status/
replication

    replicationInterval: 30

```

4. Set the IP/FQDN list of the mate sites `ingressgateway-prov` services in the following configuration under `nudr_dbcr_auditor_service` section. Enable the `dbConflictResolutionEnabled` parameter in the `global` section of the `ocudr-custom-values.yaml` file. For more information about the configuration, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation Guide*. .

```

global:
  dbConflictResolutionEnabled: "true"

nudr_dbcr_auditor_service:
  # Provide MateSite IGW IP List, Comma separated values. Provide fqdn or
  IP with port
  mateSitesIgWIPList: 'http://ocudr-ingressgateway-prov.myudr2:80,http://
ocudr-ingressgateway-prov.myudr3:80'

```

5. Perform a helm upgrade of UDR using the `helm upgrade` command. This enables database conflict resolution auditor service and you must perform the required configurations on `appinfo` service.

6. Use REST API to activate the UDR database conflict resolution feature on all the sites. The following configuration under global section of the ocudr-custom values.yaml file is used for this feature:

```
curl -v -X PUT -i 'http://ocudr-nudr-config.<site-namespace>:5001/nudr-config/v1/udr.global.cfg/GLOBAL'
-d
'{"dbServiceName":"mysql-connectivity-
service.sitea","udrServices":"All","udsfEnabled":false,"ingressHttpPort":"","
,"ingressHttpsPort":"","snssai":"2-
FFFFFF","dnn":"dnn1","autoCreate":true,"autoEnrolOnSignalling":true,"etagEn
abled":false,"version":"v3","sbiCorrelationInfoEnable":false,"consumerNF":"
PCF,UDM,NEF","nfInstanceId":"5a7bd676-ceed-44bb-95e0-
f6a55a328b03","vsaDefaultBillingDay":1,"subscriberActivityEnabled":false,"a
ddDefaultBillingDay":true,"enableControlledShutdown":false,"keyType":"msisd
n","keyRange":"1013020001-1013020002","dbConflictResolutionEnabled":true,"s
ubscriberIdentifiers":{"msisdn":[],"imsi":[],"nai":[],"extid":[]}}'
-H "Content-Type: application/json"
```

7. You must restart all the ndbapppmysqld and ndbmysqld pods on the cnDBTier deployment to activate the feature on MYSQL.

Note

Steps 2 to 5 must be performed on all UDR sites during deployment. Step 6 must be performed on all cnDBTier sites.

Perform the following steps during UDR, SLF, or EIR rollback:

1. Disable the conflict resolution, see [To Disable](#).
2. Perform a Helm rollback for cnDBTier and then NF.

To Disable

Perform the following steps to disable the conflict resolution:

1. You can disable the database Conflict Resolution feature on all the sites using the REST API. A new configuration, dbConflictResolutionEnabled, under global section has been introduced for this.

URI and Payload Sample: `curl -v -X PUT -i 'http://ocudr-nudr-config.<site-namespace>:5001/nudr-config/v1/udr.global.cfg/GLOBAL`

```
-d '{"dbServiceName":"mysql-connectivity-
service.sitea","udrServices":"All","udsfEnabled":false,"ingressHttpPort":"","
,"ingressHttpsPort":"","snssai":"2-
FFFFFF","dnn":"dnn1","autoCreate":true,"autoEnrolOnSignalling":true,"etagEn
abled":false,"version":"v3","sbiCorrelationInfoEnable":false,"consumerNF":"
PCF,UDM,NEF","nfInstanceId":"5a7bd676-ceed-44bb-95e0-
f6a55a328b03","vsaDefaultBillingDay":1,"subscriberActivityEnabled":false,"a
ddDefaultBillingDay":true,"enableControlledShutdown":false,"keyType":"msisd
n","keyRange":"1013020001-1013020002","dbConflictResolutionEnabled":false,"
subscriberIdentifiers":{"msisdn":[],"imsi":[],"nai":[],"extid":[]}}' -H
"Content-Type: application/json"
```

2. Disable the `dbConflictResolutionEnabled` parameter in the global section of the `ocudr-custom-values.yaml` file and upgrade the NF. You can ignore this step for rollback, since this feature is not available in the previous release.
3. Run the SQL command on one of the `ndbappmysqld` nodes on one of the sites. You must run the SQL command to the `ndbappmysqld` pod by logging in to the SQL terminal.

Note

In case of a two or three site setup, perform this step only on site one, on one of the `ndbappmysqld` nodes, preferably `ndbappmysqld-0` node.

- a. Run the following command to log in to the SQL terminal.

```
"mysql -h127.0.0.1 -u<udruser> -p<udrpassword">
```

- b. Run the following SQL command on one of the `ndbappmysqld` nodes on one of the sites.

```
"delete from mysql.ndb_replication where  
db='<udr_subscriber_database_name>';"
```

Note

For PCF data, the sm-data components such as `umdata` or `umdata` limits modifies individually during replication down time. The conflict resolution for the data happens for the complete sm-data based on the latest time stamp. This is applicable for 4G attributes such as quota or dynamic quota.

Database Conflict Resolution (DBCR) Service Logging Info

Debug logs are written when exceptions are cleared from data tables on SLF, UDR, or EIR. There will be one log per exception table. Info logs are written when specific IDs are cleared from the `IDXTODATA` tables and data tables during audit operation.

2.16 Network Function Scoring for a Site

The Network Function (NF) Scoring feature calculates the score for a site based on NF specific factors such as alerts. This feature determines the health of a site as compared to other sites and allows you to select an appropriate site by comparing the NF scores within or across multiple sites.

One of the use cases is the Controlled Shutdown feature that allows the operator to partially or completely isolate the site. The NF Scoring feature helps the operators to select which site to partially or completely isolate based on the NF scoring.

App-Info service queries and calculates the NF score based on the site information.

App-Info Scoring Mechanism: App-Info service reads the configurations from the common config server to check if NF Scoring functionality is enabled. It works in the following ways:

- **Continuous NF Score Calculation:** When you enable the NF Scoring feature, app-info periodically reads the configurations to calculate the score.

- **On-Demand NF Score Calculation:** When you enable the NF Scoring feature, app-info fetches all the factors or criteria to calculate the NF Score. It fetches the factors in real-time and calculates the NF score on demand.

The following factors are considered for NF scoring:

Table 2-14 NF Scoring Criteria

Factors	Default Score	Formula to Calculate Factor Score	Details
TPS	20	$\min(\text{Current-TPS} / \text{Max-TPS} * \text{Max-TPS-Score}, \text{Max-TPS-Score})$	Current-TPS includes both ingressgateway-sig and ingressgateway-prov. Max-TPS specifies the maximum TPS. Max-TPS-Score specifies the maximum score of the TPS.
Service	30	$A / N * \text{SVC-Score}$	A specifies number of available services N specifies number of configured services Max-SVC-Score specifies the maximum score of the Service Health.
Connection	20	$\min(\text{Site-Current} / \text{Site-Total} * \text{Replication-health Score}, \text{Replication-health Score})$	Connection specifies the number of connections accepted by IGW and DGW from the network. Site-Current specifies the available replication links for a particular site. Site-Total specifies the total number of sites in a geo-redundant cluster. Replication-health Score specifies the score for the replication-health.
Replication-health	30	$\min(\text{Site-Current} / \text{Site-Total} * \text{Site-Score}, \text{Site-Score})$	Site-Total specifies the total number of sites in a geo-redundant cluster. Site-Current specifies the available replication links for a particular site.
Locality-Preference	5	NA	The value of Locality-Preference is added for NF score calculation.
Critical-Alerts	2	$\text{CrN} * \text{Configured-Score-Critical-Alerts}$	CrN is the number of active critical alarms. Configured-Score-Critical-Alerts specifies the score configured by the user.

Table 2-14 (Cont.) NF Scoring Criteria

Factors	Default Score	Formula to Calculate Factor Score	Details
Major-Alerts	1	MaN * Configured-Score-Major-Alerts	MaN is the number of active Major alarms. Configured-Score-Major-Alerts specifies the score configured by the user.
Minor-Alerts	0	MaN * Configured-Score-Minor-Alerts	MiN is the number of active Minor alarms. Configured-Score-Minor-Alerts specifies the score configured by the user.

NF Scoring based on Custom Criteria

App-info supports custom criteria along with the supported scoring factors. The two methods to calculate the score for custom criteria are:

- **Ratio-based:** This method takes a ratio of the obtained metric value.
Formula: $\text{Minimum}(\text{calculated\_value\_from\_prometheus}/\text{maxValue} * \text{maxScore}, \text{maxScore})$.
For example, $\min((100/1000) * 20, 20) = \min(20, 10) = 10$.
- **Weightage-based:** This method performs a scalar multiplication of the obtained metric value.
Formula: $\text{weightage} * \text{calculated\_value\_from\_prometheus}$. For example, $2 * 40 = 80$.

Table 2-15 Custom Criteria

Site Score	Criteria	Operator Configuration							Runtime
-	-	Name	Metric Name (Name of the metric on which custom criteria calculation is based)	Enable (Indicates criteria is enabled or disabled)	Base/ Total (Maximum value expected for corresponding attribute)	Scoring Type	Max Score	Weightage	Formula
Add	NF Specific	Bulk Import	nudr_bulk_import_records_processed_total	true/false	20	Ratio	20	5	Minimum (calculated_value_from_prometheus / maxValu e * maxScore, maxScore)

Table 2-15 (Cont.) Custom Criteria

Site Score	Criteria	Operator Configuration							Runtime
-	-	Name	Metric Name (Name of the metric on which custom criteria calculation is based)	Enable (Indicates criteria is enabled or disabled)	Base/ Total (Maximum value expected for corresponding attribute)	Scoring Type	Max Score	Weightage	Formula
Add	NF Specific	Export Tool	export_tool_status_total	true/false	20	Ratio	20	5	Minimum (calculated_value_from_prometheus/maxValue*maxScore, maxScore)

Note

- For weightage-based calculation, the maximum score of the criteria is not considered.
- For ratio based-calculation, the weightage of the criteria is not considered.

The defaultSchema.json file changes include addition of enableCustomCriteria and customCriteria fields in JSON.

Note

enableCustomCriteria and customCriteria fields are optional.

The modified JSON schema is as follows:

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "sample config json",
  "description": "json validator",
  "type": "object",
  "properties": {
    "logging": {
```

```

        "type": "object",
        "description": "An explanation about the purpose of this
instance.",
        "properties": {
            "appLogLevel": {
                "enum": [
                    "CRITICAL",
                    "FATAL",
                    "ERROR",
                    "WARN",
                    "WARNING",
                    "INFO",
                    "DEBUG",
                    "TRACE"
                ],
                "type": "string",
                "description": "An explanation about the purpose of this
instance."
            },
            "packageLogLevel": {
                "type": "array",
                "minItems": 0,
                "maxItems": 6,
                "uniqueItems": true,
                "items": {
                    "anyOf": [
                        {
                            "type": "object",
                            "properties": {
                                "packageName": {
                                    "enum": [
                                        "namespace_runtime_info",
                                        "service_monitor",
                                        "mysql_monitor",
                                        "replication_status_monitor",
                                        "service_status_scraper",
                                        "log_level_agent"
                                    ],
                                    "type": "string",
                                    "description": "An explanation about
the purpose of this instance."
                                },
                                "logLevelForPackage": {
                                    "enum": [
                                        "CRITICAL",
                                        "FATAL",
                                        "ERROR",
                                        "WARN",
                                        "WARNING",
                                        "INFO",
                                        "DEBUG",
                                        "TRACE"
                                    ],
                                    "type": "string",
                                    "description": "An explanation about
the purpose of this instance."
                                }
                            }
                        }
                    ]
                }
            }
        }
    }
}

```

```

    },
    "required": [
        "packageName",
        "logLevelForPackage"
    ],
    "description": "An explanation about the
purpose of this instance."
    }
    ],
    },
    "description": "An explanation about the purpose of this
instance."
    }
    },
    "required": [
        "appLogLevel"
    ]
    },
    "nfScoring": {
        "type": "object",
        "description": "NF scoring configuration.",
        "properties": {
            "enableNFScoring": {
                "type": "boolean",
                "description": "flag used to enable and disable NF
Scoring feature"
            },
            "enableCustomCriteria": {
                "type": "boolean",
                "description": "flag used to enable or disable custom
criteria"
            },
            "tps": {
                "type": "object",
                "description": "TPS data.",
                "required": ["enable"],
                "properties": {
                    "enable": {
                        "type": "boolean"
                    },
                    "maxScore": {
                        "type": "integer",
                        "minimum": 1,
                        "maximum": 100,
                        "default": 20
                    },
                    "maxTps": {
                        "type": "integer",
                        "minimum": 1,
                        "maximum": 1000000
                    }
                }
            },
            "additionalProperties": false
        },
        "serviceHealth": {

```

```

    "type": "object",
    "description": "Service Health data.",
    "required": ["enable"],
    "properties": {
      "enable": {
        "type": "boolean"
      },
      "maxScore": {
        "type": "integer",
        "minimum": 1,
        "maximum": 100,
        "default": 30
      }
    },
    "additionalProperties": false
  },
  "signallingConnections": {
    "type": "object",
    "description": "Signalling Connections data.",
    "required": ["enable"],
    "properties": {
      "enable": {
        "type": "boolean"
      },
      "maxScore": {
        "type": "integer",
        "minimum": 1,
        "maximum": 100,
        "default": 20
      },
      "maxConnections": {
        "type": "integer",
        "minimum": 1,
        "maximum": 10000
      }
    },
    "additionalProperties": false
  },
  "replicationHealth": {
    "type": "object",
    "description": "Replication Health data.",
    "required": ["enable"],
    "properties": {
      "enable": {
        "type": "boolean"
      },
      "maxScore": {
        "type": "integer",
        "minimum": 1,
        "maximum": 100,
        "default": 30
      }
    },
    "additionalProperties": false
  },
  "localitySitePreference": {

```

```

    "type": "object",
    "description": "Locality and Site Preference data.",
    "required": ["enable"],
    "properties": {
      "enable": {
        "type": "boolean"
      },
      "score": {
        "type": "integer",
        "minimum": 1,
        "maximum": 100,
        "default": 5
      }
    },
    "additionalProperties": false
  },
  "activeAlert": {
    "type": "object",
    "description": "Active Alert Weightage.",
    "required": ["enable"],
    "properties": {
      "enable": {
        "type": "boolean"
      },
      "criticalAlertWeightage": {
        "type": "integer",
        "minimum": 0,
        "maximum": 100,
        "default": 2
      },
      "majorAlertWeightage": {
        "type": "integer",
        "minimum": 0,
        "maximum": 100,
        "default": 1
      },
      "minorAlertWeightage": {
        "type": "integer",
        "minimum": 0,
        "maximum": 100,
        "default": 0
      }
    },
    "additionalProperties": false
  },
  "customCriteria": {
    "type": "array",
    "description": "list of custom criterias",
    "items": {
      "type": "object",
      "description": "Data related to custom criteria",
      "properties": {
        "enable": {
          "type": "boolean"
        },
        "name": {

```

```

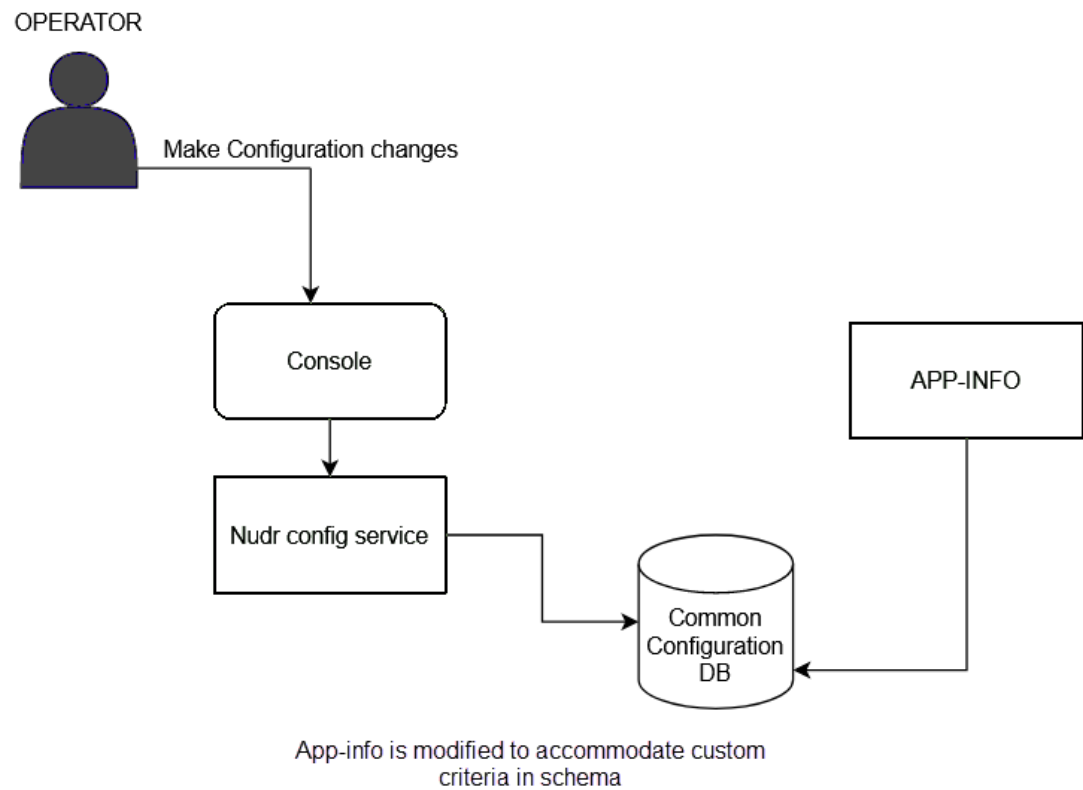
        "type": "string"
      },
      "metricName": {
        "type": "string"
      },
      "scoringType": {
        "type": "string"
      },
      "maxScore": {
        "type": "integer",
        "minimum": -1,
        "maximum": 100,
        "default": 30
      },
      "maxValue": {
        "type": "integer",
        "minimum": -1,
        "maximum": 10000
      },
      "weightage": {
        "type": "integer",
        "minimum": -1,
        "maximum": 100,
        "default": 1
      }
    }
  },
  "required": ["enable"]
},
"additionalProperties": false,
"required": [
  "enableNFScoring",
  "tps",
  "serviceHealth",
  "signallingConnections",
  "replicationHealth",
  "localitySitePreference",
  "activeAlert"
]
}
}
}

```

Formula for NF scoring of a site: Sum of TPS-Score, SVC-Score, Conn-Score, Replication-health, and Locality-Preference score subtracted from Alerts scores.

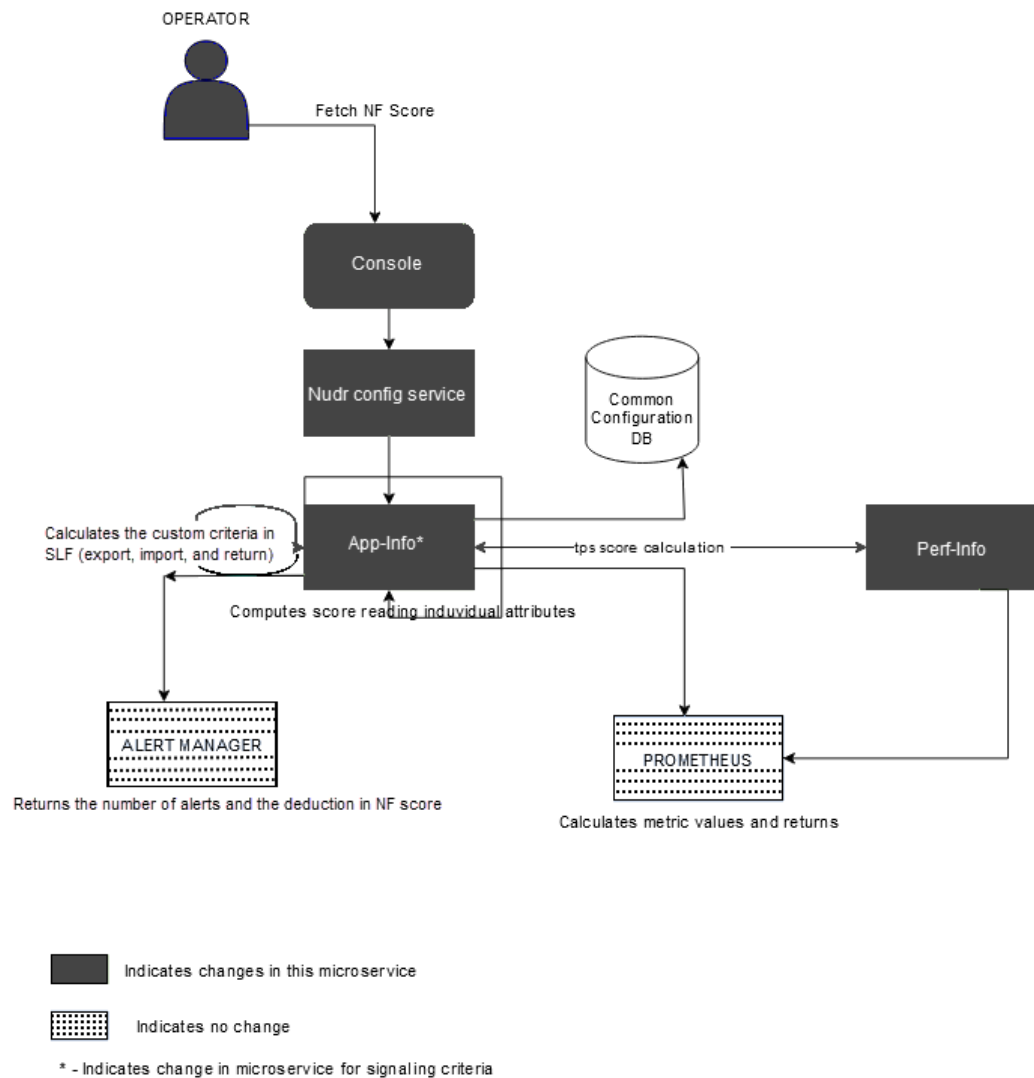
Call Flow

This section describes call flows for the NF Scoring feature:

Figure 2-7 NF Scoring Configuration

Operator use APIs to make configuration changes such as enable or disable the feature, set weightage, and parameters, which is then used by app-info.

Figure 2-8 NF Scoring



The GET NF score API and Nudr config service interacts with app-info which in turn interacts with perf-info and prometheus to calculate and return overall NF score.

Enhancements in App-Info and Perf-Info for the Signalling Connections and Transaction Per Second (TPS) Criteria

To fetch an overall score for Ingress Gateway for TPS score calculation, perf-info is modified to support changes in ingress gateway provisioning and signaling.

To fetch an overall score for Ingress Gateway for signaling score calculation, app-info is modified to support changes in ingress gateway provisioning and signaling.

You can enable the signaling connections and TPS criteria enhancements in app-info and perf-info by setting the `multipleIgwEnabled` parameter to true in the global section of the `ocudr-custom-values.yaml` file.

```
multipleIgwEnabled: true
```

The following configurations are added in the app-info and perf-info deployment.yaml file:

```
- name: MULTIPLE_IGW_ENABLED
  value: {{ .Values.global.multipleIgwEnabled | quote }}
```

The output of NF Scoring for signaling connections and TPS criteria is modified as below:

```
{
  "scoringFactors": [
    {
      "additionalInfo": {
        "actualValue": 29.95,
        "configuredValue": 33,
        "igw_prov": 19.95,
        "igw_sig": 10.0
      },
      "calculatedScore": 18.151515151515152,
      "criteria": "tps",
      "maxScore": 20
    },
    {
      "additionalInfo": {
        "actualValue": 17.0,
        "configuredValue": 40,
        "igw_prov": 12,
        "igw_sig": 4
      },
      "calculatedScore": 8.5,
      "criteria": "signallingConnections",
      "maxScore": 20
    }
  ]
}
```

Configure using CNC Console:

Enable

You can enable this feature by selecting the **Enable NF Scoring** field in the **Settings page** of NF Scoring. For more information about enabling the feature through CNC Console, see [NF Scoring Configurations](#).

Configure

The NF Scoring feature can be configured through CNC Console. For more information, see [NF Scoring Configurations](#).

Configure using REST API:

You can enable and disable the NF Scoring using REST API:

- You can perform the GET and PUT operations on the `http://<nudr-config-host>:<nudr-config-port>/udr/nf-common-component/v1/app-info/`

nfScoring to enable and disable NF Scoring and set the weightage, maximum score, and maximum value for each criteria:

```
{
  "enableNFScoring": true,
  "tps": {
    "enable": true,
    "maxScore": 20,
    "maxTps": 33
  },
  "serviceHealth": {
    "enable": true,
    "maxScore": 30
  },
  "signallingConnections": {
    "enable": true,
    "maxScore": 20,
    "maxConnections": 40
  },
  "replicationHealth": {
    "enable": true,
    "maxScore": 30
  },
  "localitySitePreference": {
    "enable": true,
    "score": 5
  },
  "activeAlert": {
    "enable": true,
    "criticalAlertWeightage": 2,
    "majorAlertWeightage": 1,
    "minorAlertWeightage": 0
  },
  "customCriteria": [{
    "name": "BulkImport",
    "enable": true,
    "metricName": "nudur_bulk_import_records_processed_total",
    "scoringType": "ratio",
    "maxScore": 20,
    "maxValue": 20,
    "weightage": 5
  }, {
    "name": "BulkExport",
    "enable": true,
    "metricName": "export_tool_status_total",
    "scoringType": "ratio",
    "maxScore": 20,
    "maxValue": 20,
    "weightage": 5
  }]
}
```

The below table provides NF Scoring settings for SLF:

Table 2-16 NF Scoring Settings for SLF

Field Name	Description
Enable NF Scoring	Specifies whether to enable or disable the NF Scoring. This is a mandatory parameter. Data Type: Boolean Default Value: false Range: true or false
TPS	Specifies the Transaction Per Second.
Enable	Enables the TPS. Data Type: Boolean Default Value: false Range: true or false
Max Score	Specifies the maximum score of the TPS. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 20 Range: 1 to 100.
Max TPS	Specifies the maximum TPS. If enable is set to true then this is a mandatory parameter. Data Type: Integer Range: 1 to 1000000
Service Health	Specifies the Service Health of a site.
Enable	Enables the Service Health. This is a mandatory parameter. Data Type: Boolean Default Value: false Range: true or false
Max Score	Specifies the maximum score of the Service Health. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 30 Range: 1 to 100
Signaling Connections	Specifies the Signaling Connections of a site.
Enable	Enables the Signaling Connections. This is a mandatory parameter. Data Type: Boolean Default Value: false Range: true or false
Max Score	Specifies the maximum score of the Signaling Connections. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 20 Range: 1 to 100
Max Connections	Specifies the maximum connections. If enable is set to true then this is a mandatory parameter. Data Type: Integer Range: 1 to 100
Replication Health	Specifies the Replication Health of a site.

Table 2-16 (Cont.) NF Scoring Settings for SLF

Field Name	Description
Enable	Enables the Replication Health. Data Type: Boolean Default Value: false Range: true or false
Max Score	Specifies the maximum score of the Replication Health. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 30 Range: 1 to 100
Locality or Site preference	Specifies the Locality or Site Preference.
Enable	Enables the Locality or Site Preference. Data Type: Boolean Default Value: false Range: true or false
Score	Specifies the score of the Locality or Site Preference. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 5 Range: 1 to 100
Active Alert	Specifies the Active Alerts of a site.
Enable	Enables the Active Alert. Data Type: Boolean Default Value: false Range: true or false
Critical Alert Weightage	The site with more critical alerts is unhealthy. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 2 Range: 0 to 100
Major Alert Weightage	The site with more major alerts is unhealthy. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 1 Range: 0 to 100
Minor Alert Weightage	The site with more minor alerts is unhealthy. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 0 Range: 0 to 100
Custom Criteria	Specifies the Custom Criteria of a site.
Enable	Specifies whether to enable or disable the Custom Criteria. Data Type: Boolean Default Value: false Range: true or false
Name	Name of the Custom Criteria. The two applicable criteria are Subscriber Bulk Import tool and Subscriber Export tool. Data Type: String

Table 2-16 (Cont.) NF Scoring Settings for SLF

Field Name	Description
Metric Name	Specifies the name of the Metric. Data Type: String
Scoring Type	Specifies the scoring type. The two scoring types are ratio and weightage. Data Type: String Default Scoring Type: ratio
Max Score	Specifies the maximum score. Data Type: Integer Default Value: 30 Range: 1 to 100
Max Value	Specifies the maximum value. Data Type: Integer Range: 1 to 10000
Weightage	Specifies the weightage of the alert. Data Type: Integer Default Value: 1 Range: 0 to 100 Note: If the scoring type is ratio the weightage is ignored.

- You can perform the GET operation on the `http://<nudr-config-host>:<nudr-config-port>/udr/nf-common-component/v1/nfscore` to fetch overall NF Score and calculated score for each criteria:

```
{
  "NFScore": 96.65151515151516,
  "detailedReport": {
    "failedFactors": [
      {
        "criteria": "activeAlert",
        "detail": "failed to calculate activeAlert score"
      }
    ],
    "scoringFactors": [
      {
        "additionalInfo": {
          "actualValue": 29.95,
          "configuredValue": 33,
          "igw_prov": 19.95,
          "igw_sig": 10.0
        },
        "calculatedScore": 18.151515151515152,
        "criteria": "tps",
        "maxScore": 20
      },
      {
        "additionalInfo": {
          "totalSvcs": 21,
          "upSvcs": 21
        },
        "calculatedScore": 30.0,

```

```

        "criteria": "serviceHealth",
        "maxScore": 30
    },
    {
        "additionalInfo": {
            "actualValue": 17.0,
            "configuredValue": 40,
            "igw_prov": 12,
            "igw_sig": 4
        },
        "calculatedScore": 8.5,
        "criteria": "signallingConnections",
        "maxScore": 20
    },
    {
        "additionalInfo": {
            "availableLinks": 0.0,
            "totalSites": 0.0
        },
        "calculatedScore": 0.0,
        "criteria": "replicationHealth",
        "maxScore": 30
    },
    {
        "calculatedScore": 5.0,
        "criteria": "localityPreference"
    },
    {
        "calculatedScore": 20,
        "criteria": "BulkImport",
        "maxScore": 20
    },
    {
        "calculatedScore": 15,
        "criteria": "BulkExport",
        "weightage": 5
    }
}
]
}

```

Table 2-17 Description of Parameters

Field Name	Mandatory (M)/ Optional(O)/ Conditional(C)	Data type	Description
NF Score	M	Float	The score of a site based on NF specific factors (such as alerts).
detailedReport	M	NA	Detailed report of NF Score calculation
scoringFactors	M	NA	Includes the factors that contributed to the NF Score

Table 2-17 (Cont.) Description of Parameters

Field Name	Mandatory (M)/ Optional(O)/ Conditional(C)	Data type	Description
criteria	C	NA	Factors involved in NF Score calculation Criteria are present for a specific factor, if it is enabled for NF Scoring.
maxScore	C	Integer	Maximum score of that factor maxScore is present for a specific factor, if it is enabled for NF Scoring.
calculatedScore	C	Float	Calculated score of that factor calculatedScore are present for a specific factor, if it is enabled for NF Scoring.
additionalInfo	O	NA	Additional information related to factor's score calculation.
configuredValue	C	Integer	An expected value configured for a factor. configuredValue is present for a specific factor, if it is enabled for NF Scoring.
actualValue	C	Float	An actual value for a particular factor after doing factor-specific configuration. actualValue is present for a specific factor, if it is enabled for NF Scoring.
totalSvcs	C	Integer	Total configured services. totalSvcs is present when the factor SVC-Health is enabled.
upSvcs	C	Integer	Total running services. upSvcs is present when the factor SVC-Health is enabled.
totalSites	C	Integer	Total configured sites totalSites is present when the factor replicationHealth is enabled.

Table 2-17 (Cont.) Description of Parameters

Field Name	Mandatory (M)/ Optional(O)/ Conditional(C)	Data type	Description
availableLinks	C	Integer	Total healthy sites. availableLinks is present when the factor replicationHealth is enabled.
critical	C	Integer	Critical alert's contribution to alert's calculated score. This attribute is present when ActiveAlerts are enabled as NF Scoring factor.
major	C	Float	Major alert's contribution to alert's calculated score. This attribute is present when ActiveAlerts are enabled as NF Scoring factor.
minor	C	Float	Minor alert's contribution to alert's calculated score. This attribute is present when ActiveAlerts are enabled as NF Scoring factor.
failedFactors	C	Float	Enabled factors that failed to contribute to NF Score. This attribute is present when an application fails to calculate the score for 1 or more factors.
detail	C	NA	Factor's failure reason. This attribute is present when there are 1 or more failed factors.

Below is an example of NF Scoring in SLF deployment:

Table 2-18 SLF Deployment NF Scoring for Site 1

Scoring Factors	Site 1			
-	Configured Value	Calculated Value	Maximum Score	Calculated Score
TPS	33	20	20	12.12
Connections	40	20	20	10
SVCs	21	21	30	30

Table 2-18 (Cont.) SLF Deployment NF Scoring for Site 1

Scoring Factors	Site 1			
-	Configured Value	Calculated Value	Maximum Score	Calculated Score
Replication	30	2	30	2
Custom Criteria Bulk Import	20	405	20	20
Custom Criteria Bulk Export	20	2	20	2
Site Preference	5	-		5
Alerts	2	2		4
Final NF score calculated	-	-	-	77.12

Table 2-19 SLF Deployment NF Scoring for Site 2

Scoring Factors	Site 2			
-	Configured Value	Calculated Value	MaximumScore	Calculated Score
TPS	33	10	20	6.06
Connections	40	10	20	5
SVCs	21	12	30	17.14
Replication	30	2	30	2
Custom Criteria Bulk Import	20	100	20	20
Custom Criteria Bulk Export	20	4	20	4
Site Preference	10	-	-	10
Alerts	2	3	-	6
Final NF score calculated	-	-	-	58.2

The final NF score for Site 1 is **77.12** and Site 2 is **58.2**. The preferred is Site 1.

Site 1 output is given below:

```
{
  "NFScore": 77.12,
  "detailedReport": {
    "failedFactors": [],
    "scoringFactors": [{
      "additionalInfo": {
        "actualValue": 20,
        "configuredValue": 33,
        "igw_prov": 10,
        "igw_sig": 10
      },
      "calculatedScore": 12.12,
      "criteria": "tps",
      "maxScore": 20
    }
  ]
}
```

```

    {
      "additionalInfo": {
        "totalSvcs": 21,
        "upSvcs": 21
      },
      "calculatedScore": 30,
      "criteria": "serviceHealth",
      "maxScore": 30
    },
    {
      "additionalInfo": {
        "actualValue": 20,
        "configuredValue": 40,
        "igw_prov": 10,
        "igw_sig": 10
      },
      "calculatedScore": 20,
      "criteria": "signallingConnections",
      "maxScore": 20
    },
    {
      "additionalInfo": {
        "availableLinks": 2,
        "totalSites": 30
      },
      "calculatedScore": 2,
      "criteria": "replicationHealth",
      "maxScore": 30
    },
    {
      "additionalInfo": {
        "critical": -4.0,
        "major": 0.0,
        "minor": 0.0
      },
      "calculatedScore": -4.0,
      "criteria": "activeAlert"
    },
    {
      "calculatedScore": 5.0,
      "criteria": "localityPreference"
    },
    {
      "calculatedScore": 20,
      "criteria": "BulkImport",
      "maxScore": 20
    },
    {
      "calculatedScore": 2,
      "criteria": "BulkExport",
      "maxScore": 20
    }
  ]
}

```

Site 2 output is given below:

```
{
  "NFScore": 58.12,
  "detailedReport": {
    "failedFactors": [],
    "scoringFactors": [{
      "additionalInfo": {
        "actualValue": 10,
        "configuredValue": 33,
        "igw_prov": 5,
        "igw_sig": 5
      },
      "calculatedScore": 6.06,
      "criteria": "tps",
      "maxScore": 20
    },
    {
      "additionalInfo": {
        "totalSvcs": 21,
        "upSvcs": 12
      },
      "calculatedScore": 17.14,
      "criteria": "serviceHealth",
      "maxScore": 30
    },
    {
      "additionalInfo": {
        "actualValue": 10,
        "configuredValue": 40,
        "igw_prov": 5,
        "igw_sig": 5
      },
      "calculatedScore": 5,
      "criteria": "signallingConnections",
      "maxScore": 20
    },
    {
      "additionalInfo": {
        "availableLinks": 2,
        "totalSites": 30
      },
      "calculatedScore": 2,
      "criteria": "replicationHealth",
      "maxScore": 30
    },
    {
      "additionalInfo": {
        "critical": -6.0,
        "major": 0.0,
        "minor": 0.0
      },
      "calculatedScore": -6.0,
      "criteria": "activeAlert"
    }
  ]
}
```

```

        "calculatedScore": 10.0,
        "criteria": "localityPreference"
    },
    {
        "calculatedScore": 20,
        "criteria": "BulkImport",
        "maxScore": 20
    },
    {
        "calculatedScore": 4,
        "criteria": "BulkExport",
        "maxScore": 20
    }
  ]
}

```

To Observe

The following alert is introduced for NF Scoring feature:

- NFScoreCalculationFailed

See UDR [Alert Details](#) section for more information alerts.

2.17 Converged Quota Support

This section describes the Converged Quota Support feature.

2.17.1 Converged Quota Support for UDR

This feature updates the data model to support storage of 4G policy quota and dynamic quota under sm-data/umData and sm-data/umDataLimits respectively from VSA. This feature enables cnPCRF to access and manage 4G and 5G quota from umData and umDataLimits. The schema management APIs interface is updated to include the default 4G policy quota attributes in umData and umDataLimits. The following interfaces are compatible with the converged data model:

- Provisioning interface using SOAP/XML
- Bulk provisioning using file import in iXML and CSV format. For more information, see [Bulk Import Operations in iXML Format](#) and [Bulk Import Operations in CSV Format](#).
- N36 signaling interface. For more information, see [Policy N36 APIs](#).
- Provisioning Interface using REST/JSON. For more information, see [Provisioning Using REST/JSON](#).
- On-Demand Migration. For more information, see [On-Demand Migration](#).
- Diameter-Sh. For more information, see [Diameter-Sh](#).
- Migration Tool. For more information, see [Migrating Subscriber Data from 4G UDR to 5G UDR](#).

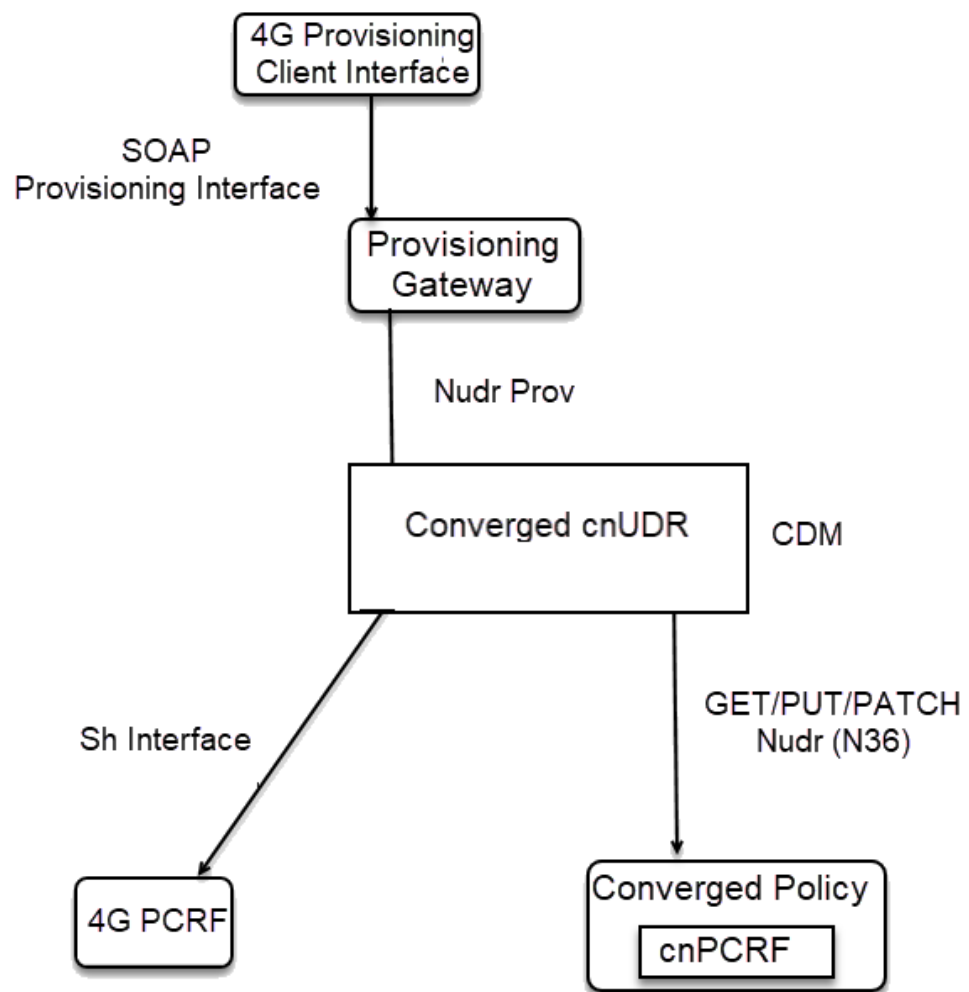
Subscribers provisioned using a UDR deployment in 23.1.x release or older release version before the introduction of Converged Quota feature will remain with the old quota model. Upgrading UDR to 23.2.x or later releases does not modify the quota model for the existing

subscribers in the database. You must migrate the subscribers to the new quota model by updating the policy data using the following interfaces:

- Bulk provisioning using file import in iXML and CSV format.
- Provisioning APIs using SOAP and REST
- N36 signaling operations from PCF
- Migration Tool

cnPCRF interacts with 5G UDR for both 4G and 5G signaling data. cnPCRF supports only the N36 interface toward 5G UDR for 4G call flows. The cnUDR must store the quota information under sm-data/umData and sm-data/umDataLimits.

Figure 2-9 Converged Quota Support



Using Provisioning Gateway, 4G provisioning system provisions 4G PCRF data over SOAP/XML interface on 5G UDR. Provisioning Gateway converts SOAP/XML to REST/JSON and forward to 5G UDR. 5G UDR converges the 4G policy quota and dynamic quota data to 5G data model and stores in 5G UDR. When 4G PCRF accesses this data in 5G UDR using

Diameter-Sh interface, 5G UDR converts the 5G data to 4G data and respond through the Diameter-Sh interface.

The UDR provides Migration tool, On-Demand Migration and Diameter-Sh support for data transition from 4G to 5G UDR. For more information, see [Migrating Subscriber Data from 4G UDR to 5G UDR](#), [On-Demand Migration](#), and [Diameter-Sh](#).

Migration Tool

The migration tool is an external tool that is used to migrate the subscriber data from source UDR (4G OCUDR) to target UDR (5G UDR). Migration tool is delivered as a separate Helm chart and you must install it manually. The migration tool supports migration of the subscribers in ranges, based on the configuration provided in the Helm chart. Migration tool supports the converged quota feature by default. For more information, see [Migration Tool](#).

Diameter-Sh

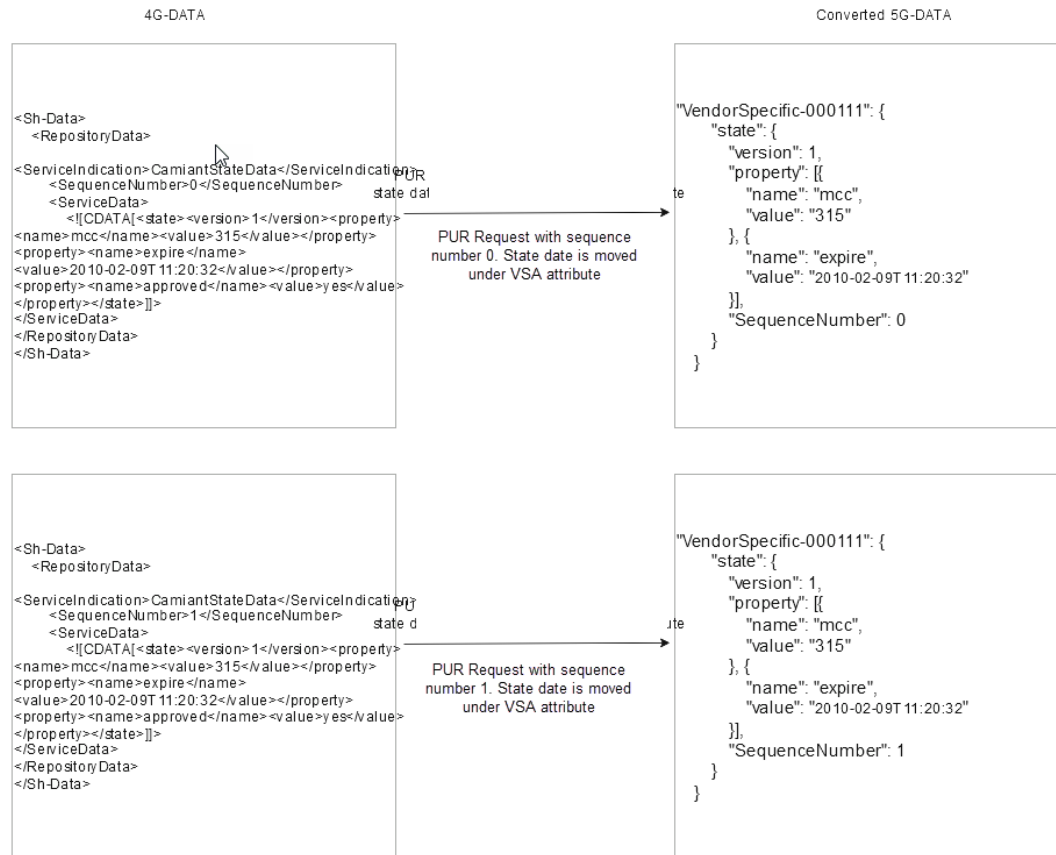
For Diameter-Sh to be compatible with the updated data model, the following call flows are updated:

- Sh-User Data Request (UDR)
- Sh-Profile Update Request (PUR)
- Sh-Subscribe Notification Request (SNR)
- Sh- Push Notification Request (PNR)

In the converged data model, ETag (Entity Tag) value is used to update quota and dynamic quota and sequence number is used to update subscriber and state entities. Sequence number is used on Diameter-Sh and ETag is used for SBI interface.

Sequence Number

5G UDR uses sequence number for subscriber and state entities that are present under VSA attributes. Sequence number increments every time the subscriber or state entities are updated.



E-Tag

UDR uses ETag value for quota and dynamic quota update, which are present in sm-data. ETag value increments every time quota or dynamic quota is updated as follows:

- ETag value is 0, if auto create is enabled by default and quota or dynamic quota is created using PUR, SNR, or UDR request.
- When PUR request is sent, the sequence number in the request validates against the ETag value (Example 0). The ETag value is incremented by 1 for each quota or dynamic quota update.
- If two different sequence numbers (Example 2 and 3) are sent during PUR request, the sequence number with the highest value (Example 3) is selected to validate against the ETag value. ETag value increments accordingly.

Diameter-Sh request only sends the sequence number to 5G UDR. 5G UDR updates the versions using the ETag functionality. ETag value is used to update versions for both Diameter-Sh and N36 requests.

The below table provides quota mapping of 4G UDR to 5G UDR attributes and data type:

Table 2-20 Quota Mapping from 4G UDR to 5G UDR

4G UDR Attribute	5G UDR Parent JSON Attribute	5G UDR JSON Attribute
name Data Type: String	umData	name Data Type: String

Table 2-20 (Cont.) Quota Mapping from 4G UDR to 5G UDR

4G UDR Attribute	5G UDR Parent JSON Attribute	5G UDR JSON Attribute
cid Data Type: String	umData	limitId Data Type: String
time Data Type: String	umData/customAllowedUsage	duration Data Type: DurationSec
totalVolume Data Type: String	umData/customAllowedUsage	totalVolume Data Type: Volume
inputVolume Data Type: String	umData/customAllowedUsage	downlinkVolume Data Type: Volume
outputVolume Data Type: String	umData/customAllowedUsage	uplinkVolume Data Type: Volume
serviceSpecific Data Type: String	umData	serviceSpecific Data Type: Volume
nextResetTime Data Type: String	umData	resetTime Data Type: DateTime
Type Data Type: String	umData	type Data Type: string
grantedTotalVolume Data Type: String	umData/grantedUsage	totalVolume Data Type: Volume
grantedInputVolume Data Type: String	umData/grantedUsage	downlinkVolume Data Type: Volume
grantedOutputVolume Data Type: String	umData/grantedUsage	uplinkVolume Data Type: Volume
grantedTime Data Type: String	umData/grantedUsage	duration Data Type: DurationSec
grantedServiceSpecific Data Type: String	umData	grantedServiceSpecific Data Type: string
QuotaState Data Type: String	umData	quotaState Data Type: string
RefInstanceId Data Type: String	umData	refInstanceId Data Type: string
Custom 1-n Data Type: Data type entered in schema	umData	custom 1-n Data Type: Data type entered in schema

The below table provides dynamic quota mapping of 4G UDR to 5G UDR attributes and data type:

Table 2-21 Quota Mapping from 4G UDR to 5G UDR

4G UDR Attribute	5G UDR Parent JSON Attribute	5G UDR JSON Attribute
Type Data Type: String	umDataLimit	type Data Type: String
name Data Type: String	umDataLimit	name Data Type: String

Table 2-21 (Cont.) Quota Mapping from 4G UDR to 5G UDR

4G UDR Attribute	5G UDR Parent JSON Attribute	5G UDR JSON Attribute
InstanceId Data Type: String	umDataLimit	limitId Data Type: String
Priority Data Type: String	umDataLimit	priority Data Type: String
InitialTime Data Type: String	umDataLimit/usageLimit	duration Data Type: DurationSec
InitialTotalVolume Data Type: String	umDataLimit/usageLimit	totalVolume Data Type: Volume
InitialInputVolume Data Type: String	umDataLimit/usageLimit	downlinkVolume Data Type: Volume
InitialOutputVolume Data Type: String	umDataLimit/usageLimit	uplinkVolume Data Type: Volume
InitialServiceSpecific Data Type: String	umDataLimit	initialServiceSpecific Data Type: Volume
activationdatetime Data Type: String	umDataLimit	activationdatetime Data Type: DateTime
expirationdatetime Data Type: String	umDataLimit	endDate Data Type: DateTime
purchasedatetime Data Type: String	umDataLimit	startDate Data Type: DateTime
Duration Data Type: String	umDataLimit	validityDuration Data Type: DurationSec
InterimReportingInterval Data Type: String	umDataLimit	interimReportingInterval Data Type: DurationSec
Custom 1-n Data Type: Data type entered in schema	umDataLimit	custom 1-n Data Type: Data type entered in schema

Mapping Custom Attributes

You must complete the following configurations on UDR and Provisioning Gateway for custom attributes added in the quota or dynamic quota:

- Update the attribute mapping using management APIs. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*
- Update the Subscriber Entity Configuration (SEC) for SOAP/XML interface field validation. For more information, see *Oracle Communications Cloud Native Core, Provisioning Gateway Guide* and *Oracle Communications Cloud Native Core, Provisioning Gateway Interface Specification Guide*.
- Update the sm-data schema. For more information, see "PCF Schema Management" in *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

You must complete the following configurations on UDR and Provisioning Gateway for custom attributes under the state and subscriber entity:

- Update the SEC for SOAP/XML interface field validation. For more information, see *Oracle Communications Cloud Native Core, Provisioning Gateway Guide* and *Oracle Communications Cloud Native Core, Provisioning Gateway Interface Specification Guide*.
- Update the VSA schema. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide* and *Oracle Communications Cloud Native Core, Provisioning Gateway Interface Specification Guide*.

Note

If the custom fields (under UM-data/UM-dataLimits or subscriber profile/state) are added but not defined in the sm-data schema, UDR accepts the custom fields received in the request payload and skips the schema validation for custom fields.

Configuration to Add Quota and Dynamic Quota Version

Quota and dynamic quota version are static values. You can configure the values in the global section of the custom-values.yaml file during the UDR deployment. The configuration is as follows:

```
#Quota version
quotaVersion: 3
```

```
#Dynamic quota version
dynamicQuotaVersion: 3
```

Adding SEC on Provisioning Gateway

To accept and validate new custom entries, you must add the custom entries to the SEC configuration on provisioning gateway. For more information about adding a new custom entity, see "Subscriber Entity Configuration (SEC)" in *Oracle Communications Cloud Native Core, Provisioning Gateway Guide*.

Adding Schema and Attribute Mapping Table

For the newly added custom entities, must add attribute mapping from 4G attributes to 5G attributes. If the custom entities are not added in the attribute mapping, UDR ignores the custom entities during provisioning. For more information about adding attribute mapping, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Note

For Multiple site deployment, attribute mapping updates using Management APIs must done on each site.

Update sm-data Schema for Validation

Validate the added custom entities against schema after the 4G data is converged to 5G data. For more information, see "PCF Schema Management" in *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Provisioning Using SOAP/XML

There are no changes to the provisioning APIs. For more information, see *Oracle Communications Cloud Native Core, Provisioning Gateway Interface Specification Guide*.

Note

SOAP/XML APIs are used for provisioning 4G only policy data only. SOAP/XML APIs cannot be used for 5G policy data.

Provisioning Using REST/JSON

For provisioning 4G and 5G policy data over REST interface perform the follow steps.

- The 4G subscriber profile fields and state must be under sm-data/VendorSpecific-000111
- Add all 4G quota and dynamic quota information under sm-data/umData and sm-data/umDataLimits respectively.
- All 5G policy data must be under sm-data.

For more information, see "Provisioning Operations for PCF Data" and "4G Policy Data Operations" in *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

The recommended way of provisioning 4G and 5G policy data is as follows:

Operation Type	Resource URI
PUT	{apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}
GET	{apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}
DELETE	{apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}

The below mentioned APIs are not changed and are used for quota and dynamic quota operations:

Table 2-22 REST APIs for 4G Policy Data

Operation Type	Resource Name	REST URI
PUT, GET	Update complete 4G policy data or update individual entities Retrieve complete 4G policy data	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa
PUT	Update quota row operations	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/usage/quota/row
GET	Retrieve quota	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa?fields=usage

Table 2-22 (Cont.) REST APIs for 4G Policy Data

Operation Type	Resource Name	REST URI
GET	Retrieve quota row	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/usage/quota/row?name=Q1&cid=11223344
GET	Retrieve quota row fields	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/usage/quota/row?fields=outputvolume&name=Q3&cid=11223344
DELETE	Delete quota	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa?fields=usage
DELETE	Delete quota row	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/usage/quota/row
PATCH	Patch quota row	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/usage/quota/row
PUT	Update dynamic quota row	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/definition/DynamicQuota/row
GET	Retrieve dynamic quota	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa?fields= definition
DELETE	Delete dynamic quota	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa?fields=definition
DELETE	Delete dynamic quota row	{ apiRoot}/nudr-dr-prov/v2/policy-data/msisdn-33123654862/sm-data/vsa/definition/DynamicQuota/row?name=DQ1
PATCH	Patch dynamic quota row	{ apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/definition/DynamicQuota/row

Table 2-22 (Cont.) REST APIs for 4G Policy Data

Operation Type	Resource Name	REST URI
GET	Retrieve dynamic quota row	{ apiRoot }/nudr-dr-prov/v2/policy-data/{ ueId }/sm-data/vsa/definition/DynamicQuota/row?name=Q1&cid=11223344
GET	Retrieve dynamic quota row field	{ apiRoot }/nudr-dr-prov/v2/policy-data/{ ueId }/sm-data/vsa/definition/DynamicQuota/row?name=DQ3&InstanceId=15678&fields=InstanceId,InitialTotalVolume,InitialInputVolume
DELETE	Delete complete 4G policy data	{ apiRoot }/nudr-dr-prov/v2/policy-data/{ ueId }/sm-data/vsa?fields=all

Provisioning Bulk Import Using iXML and CSV

For provisioning bulk import using iXML and CSV, see [Bulk Import Operations in iXML Format](#) and [Bulk Import Operations in CSV Format](#).

Policy N36 APIs

If the On-Demand Migration feature is enabled and 4G subscriber information is not available on 5G UDR, 5G UDR reads and provisions the 4G policy sm-data on 5G UDR based on request received from PCRF.

For REST APIs supported on 5G UDR N36 interface, see "Operations Supported for PCF Data" in *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Note

The APIs mentioned for provisioning operation of policy data must not have the usage and definition information under sm-data/VendorSpecific-000111 content. The usage and definition information must be converted to 5G data and placed under sm-data/umData and sm-data/umDataLimits.

2.18 Controlled Shutdown of an Instance

UDR SLF supports controlled shutdown feature to provide the partial or complete isolation of the site from the network so that the operator can perform necessary recovery procedures when required. It helps the operator to perform the recovery procedures as per the requirement.

Note

Controlled Shutdown feature is supported by UDR only on SLF mode.

The site shutting down is achieved as follows:

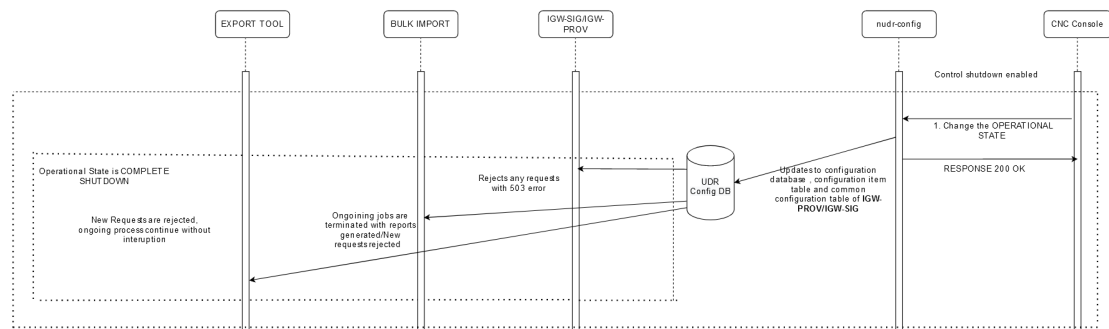
- Dropping signaling and provisioning requests on Ingressgateway-Sig and Ingressgateway-Prov with error codes.
- Updating the NF status as SUSPENDED at NRF.
- Export is not initiated in the subscriber export tool when the instance is in SHUTDOWN phase. Alert is raised to indicate this state.
- Bulk provisioning operations using subscriber bulk import tool is terminated.

Operational State

The site can be in one of the two possible operational states, NORMAL or COMPLETE SHUTDOWN. The operational state of the NF and the updates made on the operational state through REST API is internally updated in the following three services:

- config database
- Ingressgateway-sig common configuration
- Ingressgateway-prov common configuration

Figure 2-10 Operational State of the NF

**App-Info**

App-info monitors the operational state using GET API of the UDR. To support control shutdown feature the following changes is made on the app-info:

Controlled shutdown feature
 controlledshutdown:
 operationalStateResourcePath: "/operationalState"

Ephemeral Storage Request Configurations
 global:
 logStorage: *containersl_logStorageRequestsRef crictlStorage:
 *containersCrictlStorageRequestsRef
 nfType: udr
 enableControlledShutdown: *enableControlledShutDown

Ingress Gateway

Filter need to be added in the routes config section for each route id for both Ingressgateway-Sig and Ingressgateway-Prov as follows:

```
# configuring routes
routesConfig:
- id: trafficjncapping_http
  uri: http://-ff .Release.Name }}-nuds-drsevice:5001
  path: /nuds-dr/**
  order: 1
  metadata:
    # oAuthValidation route level
    oAuthValidator:
      enabled: *oAuthEnabled
    # xfcHeaderValidation route level
    xfcHeaderValidation:
      # When enabled on global section, the same gets referred here
      # Set to false, if needs to be disabled for this particular route
      validationEnabled: *xfcValidationEnabled
  filters:
    controlledShutdownFilter:
      applicableShutdownStates:
        - "COMPLETE SHUTDOWN"
```

For each route you must configure the error code using Helm or REST API. The REST API is enabled by default and the errorcode mapping for each route is done by performing a PUT operation using `v1/igw-sig/controlledshutdownerrormapping`.

Helm:

```
# Controlled shutdown feature
enableControlledShutdown: *enableControlledShutDown
controlledShutdownConfigMode: HELM
controlledShutdownErrorMapping:
  routeErrorProfileList :
    - routeId: traffic_mapping_http
      errorProfileName: "error503"
```

By default the `controlledshutdownerrormapping` URL is loaded with the SLF group URL as part of ingress gateway as shown below:

- For Ingressgateway-Sig REST API `http://10.75.229.85:30168/uds/nf-common-component/v1/igw-sig/controlledshutdownerrormapping`
`igw-sig controlledshutdownerrormapping`

```
{
  "routeErrorProfileList": [
    {
      "routeId": "traffic_mapping_http_group",
      "errorProfileName": "error503"
    }
  ]
}
```


- For Ingressgateway-Prov REST API `http://10.75.229.85:30168/udr/nf-common-component/v1/igw-prov/controlledshutdownerrormapping`
`igw-prov controlshutdownerrormapping`

```
{
  "routeErrorProfileList": [
    {
      "routeId": "traffic_mapping_http_group_prov",
      "errorProfileName": "error503"
    },
    {
      "routeId": "traffic_mapping_http_slf_group_prov",
      "errorProfileName": "error503"
    }
  ]
}
```

To Enable

You can enable or disable the control shutdown feature by setting the `enableControlledShutdown` parameter to true or false in the global section of the `ocudr-custom-values.yaml` file. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

You can enable or disable the operation state by setting the `enableControlledShutdown` parameter to true or false in the [Global Configurations](#) in the CNC Console. You can configure the operation state of a site through CNC Console. For more information, see [Operational State](#).

Configure using REST API:

You can perform the PUT operation on the `http://10.75.229.206:30240/nudr-config/v1/udr.global.cfg/GLOBAL` to configure the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "dbServiceName": "mysql-connectivity-service.occne-ndb",
  "udrServices": "All",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "sbiCorrelationInfoEnable": false,
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceed-44bb-95e0-f6a55a328b03",
  "vsaDefaultBillingDay": 1,
  "subscriberActivityEnabled": false,
  "addDefaultBillingDay": true,
  "enableControlledShutdown": true,
  "keyType": "imsi",
  "keyRange": "2013020001-2013020002",
}
```

```

    "nfType": "UDR",
    "nfFqdn": "udr001.oracle.com",
    "suppressNotificationEnabled": true,
    "configClientConnectTimeout": 1000,
    "configClientReadTimeout": 1000,
    "subscriberIdentifiers": {
      "msisdn": [],
      "imsi": [],
      "nai": [],
      "extid": []
    }
  }
}

```

SLF operational state REST API is introduced to make the application COMPLETE_SHUTDOWN or NORMAL.

Perform the PUT operation on the `http://10.75.229.85:30254/udr/nf-common-component/v1/operationalState`

opstate

```

{
  "operationalState": "NORMAL"
}

```

Perform the GET operation on the `http://10.75.229.85:30254/udr/nf-common-component/v1/operationalState`

opstte

```

{
  "operationalState": "NORMAL"
}

```

REST API OperationalHistory is introduced to get the history of last five operations that is performed on the operationalState.

Perform the GET operation on the `http://10.75.229.85:30254/udr/nf-common-component/v1/operationalStateHistory`

opstathisto

```

{
  "operationalState": "NORMAL",
  "operationalStateHistory": [
    {
      "operationalState": "NORMAL",
      "timestamp": "2023-02-02 13:42:16.621639149",
      "status": "SUCCESS"
    },
    {
      "operationalState": "COMPLETE_SHUTDOWN",
      "timestamp": "2023-02-02 13:42:08.148351775",
      "status": "SUCCESS"
    },
    {

```

```

        "operationalState": "NORMAL",
        "timestamp": "2023-02-02 13:35:07.111230791",
        "status": "SUCCESS"
      },
      {
        "operationalState": "COMPLETE_SHUTDOWN",
        "timestamp": "2023-02-02 13:33:22.06605161",
        "status": "SUCCESS"
      }
    ]
  }
}

```

Perform the GET operation on `curl -v -XGET -H "Content-type: application/json" 'http://ocudr-nudr-config:5001/udr/nf-common-component/v1/operationalStateHistory?limit=3'` to get last three operationalStateHistory.

Note

You can get maximum of five operationalStateHistory.

opstathisto

```

{
  "operationalState": "NORMAL",
  "operationalStateHistory": [
    {
      "operationalState": "NORMAL",
      "timestamp": "2023-02-02 13:42:16.621639149",
      "status": "SUCCESS"
    },
    {
      "operationalState": "COMPLETE_SHUTDOWN",
      "timestamp": "2023-02-02 13:42:08.148351775",
      "status": "SUCCESS"
    }
  ]
}

```

Note

- If controlled shutdown is enabled and if the status is COMPLETE_SHUTDOWN, the subscriber export tool rejects any new jobs that is triggered and a metric is raised for the rejection.
- If controlled shutdown is enabled and if the status is COMPLETE_SHUTDOWN, the subscriber bulk import tool terminates the ongoing jobs by generating report of the progress of the job. The jobs is marked as SUSPENDED in the configurationDB, and any new jobs that is triggered is rejected.

To Observe

The following metrics are introduced for controlled shutdown feature:

- nudr_config_operational_status

- `export_tool_controlled_shutdown`

The following alert is introduced for controlled shutdown feature:

- `nudr_config_operational_status`

See UDR [Metrics](#) and [Alert Details](#) section for more information on metrics and alerts.

2.19 Auto Enrollment and Auto Create Features

This section describes auto enrollment and auto create features on UDR.

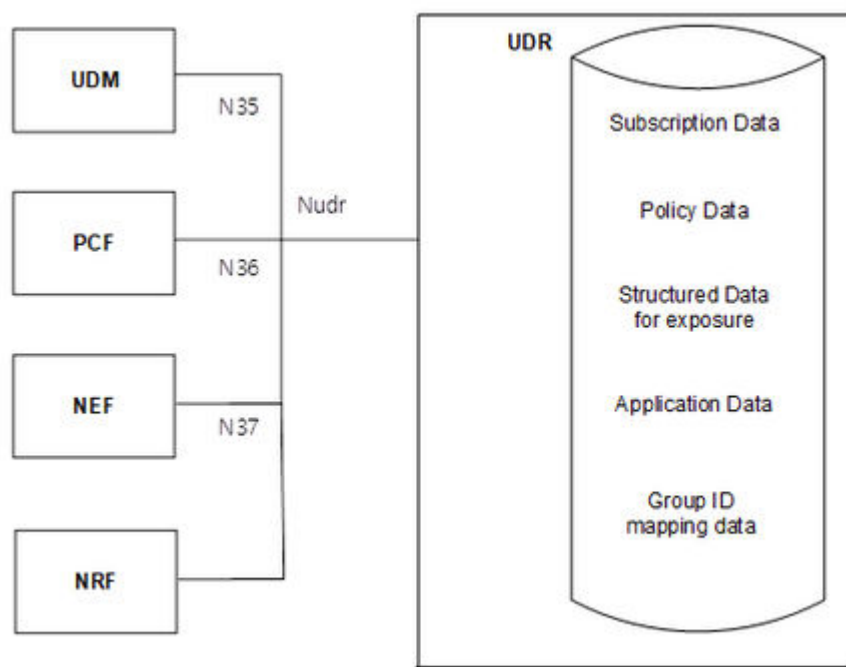
2.19.1 Auto Enrollment

UDR has various consumer NFs which store the NF specific data. The Nudr interface is used by the NF such as UDM, PCF, and NEF to access a particular set of the data that is stored in UDR database. The data must be provisioned by a provisioning system before the consumer NF can access the data. The provisioning system sends all the subscriber keys in the request as part of creating data using REST/JSON commands. Once the provisioning is successful the consumer NF can access the created subscriber record using GET or PATCH operations.

Consumer NF such as cnPCRF need UDR to enroll the subscriber record with policy data that is sent on the N36 interface. This is required for subscribers in roaming scenarios, where cnPCRF does a PATCH operation to update the subscriber data. Auto enrollment enables UDR to auto enroll the subscriber if the subscriber record does not exist.

Auto enrollment also supports UDR in creating mandatory attribute `smPolicySnssaiData` with configured default values when NFs such as cnPCRF does not include `smPolicySnssaiData` attribute in `sm-data`.

Figure 2-11 Data Storage Architecture



The auto enrollment feature in UDR supports the auto provisioning of subscribers for the following signalling APIs:

- PATCH on SessionManagementPolicyData
- PUT on UsageMonitoringInformation

The mandatory parameters in SessionManagementPolicyData such as SNSSAI data must be sent in PATCH request as per the 3GPP specifications.

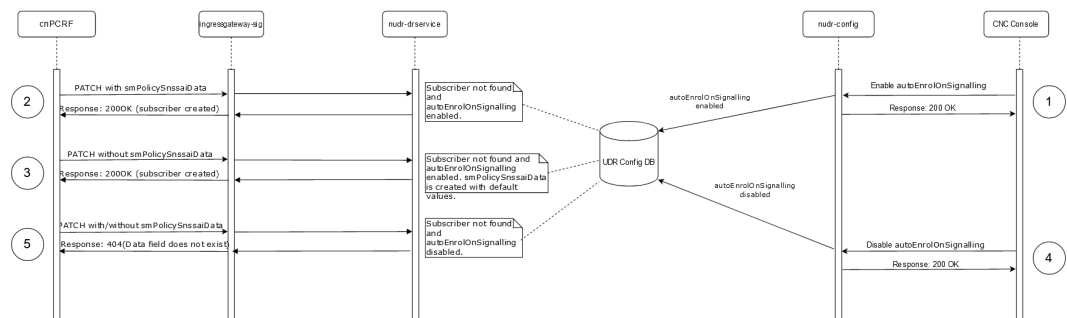
If the mandatory parameters is not available for cnPCRF to send this information or SessionManagementPolicyData resource is not present for the subscriber. UDR supports addition of mandatory attributes with default values under SessionManagementPolicyData.

You can enable or disable the auto enrollment feature by setting the autoEnrolOnSignalling parameter to true or false in the global section of the ocudr-custom-values.yaml file.

The following enhancements is made on signaling APIs PATCH and PUT operations to support the auto enrollment feature:

- The following enhancements is made for PATCH operation:
Base URI: {apiRoot}/nudr-dr/v2/policy-data/ues/{ueId}/sm-data)
 1. If subscriber is not found on UDR database and when a PATCH request is received with smPolicySnssaiData and autoEnrolOnSignalling is set to true, the subscriber is provisioned.
 If smPolicySnssaiData is not present in the request and autoEnrolOnSignalling is set to true, then smPolicySnssaiData is created with configured default values and the subscriber is provisioned.
 2. If subscriber is found on UDR database and SessionManagementPolicyData is not present. When a PATCH request without smPolicySnssaiData is received and autoEnrolOnSignalling is set to true, then smPolicySnssaiData is created with configured default values and the SessionManagementPolicyData is created.
 If smPolicySnssaiData is present in the request, SessionManagementPolicyData is created.

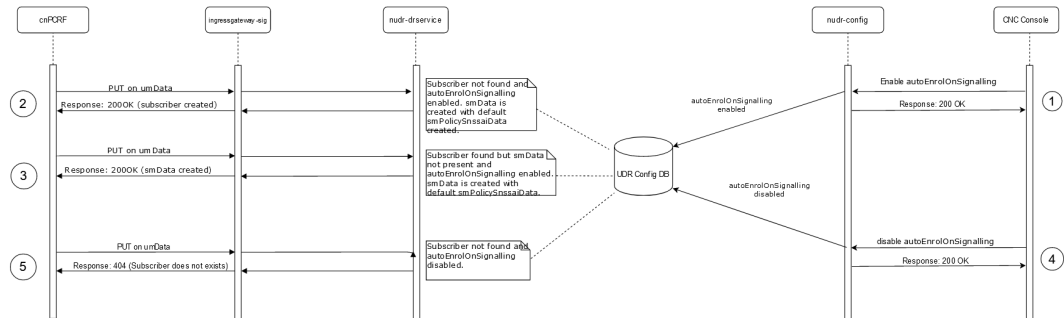
Figure 2-12 PATCH operation for Auto Enrollment



- The following enhancements is made for PUT operation:
Base URI: {apiRoot}/nudr-dr/v2/policy-data/ues/{ueId}/sm-data/{usageMonId}
 1. If subscriber is not found on UDR database, when a PUT request is received on umData and autoEnrolOnSignalling is set to true, then the subscriber is created with the given payload under umData tag of SessionManagementPolicyData along with configured default values of smPolicySnssaiData.

2. If subscriber is found on UDR database and SessionManagementPolicyData is not present, when a PUT request is received on umData and autoEnrolOnSignalling is set to true, then the SessionManagementPolicyData is created with the given payload under umData tag of SessionManagementPolicyData along with default values of smPolicySnssaiData.

Figure 2-13 PUT operation for Auto Enrollment



To Enable

You can enable or disable the auto enrollment feature by setting the `autoEnrolOnSignalling` parameter to true or false in the global section of the `ocudr-custom-values.yaml` file. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Note

This flag is applicable only for N36 signaling requests.

You can enable or disable the feature by setting the `autoEnrolOnSignalling` parameter to true or false in the [Global Configurations](#) in the CNC Console.

Configure using REST API

You can perform the PUT operation on the `http://10.75.215.124:32523/nudr-config/v1/udr.global.cfg/GLOBAL` to configure the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "dbServiceName": "mysql-connectivity-service.occne-ndb",
  "udrServices": "All",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "sbiCorrelationInfoEnable": false,
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceed-44bb-95e0-f6a55a328b03",
```

```

    "vsaDefaultBillingDay": 1,
    "subscriberActivityEnabled": false,
    "addDefaultBillingDay": true,
    "enableControlledShutdown" : true,
    "keyType": "imsi",
    "keyRange": "2013020001-2013020002",
    "nfType": "UDR",
    "nfFqdn": "udr001.oracle.com",
    "suppressNotificationEnabled": true,
    "configClientConnectTimeout": 1000,
    "configClientReadTimeout": 1000,
    "subscriberIdentifiers": {
        "msisdn": [],
        "imsi": [],
        "nai": [],
        "extid": []
    }
}

```

To Observe

The following metrics are introduced for auto enrollment feature. The metrics will count the total number of requests for which default value of smPolicySnssaiData and SNSSAI is applied.

- udr_default_smPolicySnssaiData_creation
- total_auto_enrol_requests
- udr_auto_enrol_subscriber_success

See UDR [Metrics](#) section for information on metrics.

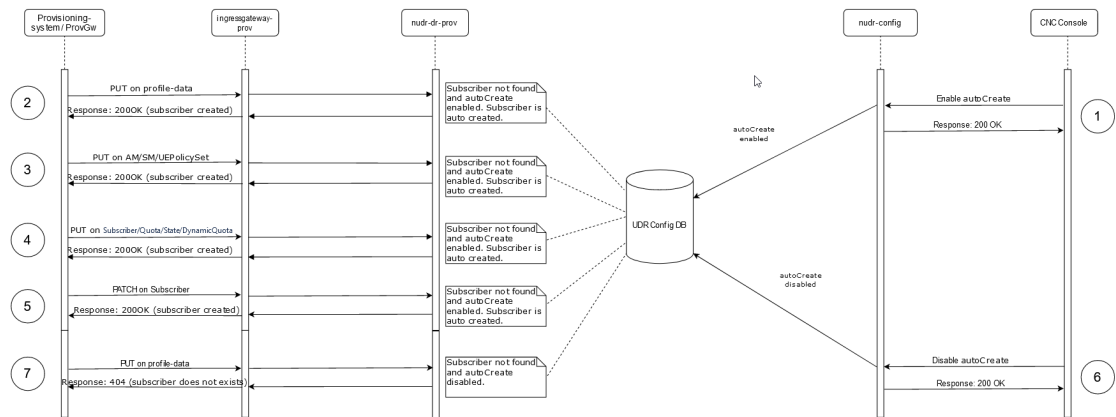
Logging

Info loggers is added when default value of smPolicySnssaiData and SNSSAI is applied for the requests.

2.19.2 Auto Create

The auto create feature is used when a provisioning system updates a part of subscriber data instead of creating the subscriber profile, keys, or the policy data. In this feature the subscriber record is auto created on UDR with the information available in the provisioning request.

The feature is applicable from Provisioning Gateway. Provisioning Gateway supports SOAP/XML interface, when the provisioning systems does a PUT operation on policy data instead of a POST operation, the PUT operation auto creates the subscriber record.

Figure 2-14 Auto Create

To Enable

You can enable or disable the auto create feature by setting the `autoCreate` parameter to `true` or `false` in the global section of the `ocudr-custom-values.yaml` file. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Note

This flag is applicable only for provisioning and diameter requests. This is the only method to enable this feature.

You can enable or disable the feature by setting the `autoCreate` parameter to `true` or `false` in the [Global Configurations](#) in the CNC Console.

Configure using REST API

You can perform the PUT operation on the `http://10.75.215.124:32523/nudr-config/v1/udr.global.cfg/GLOBAL` to configure the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```

{
  "dbServiceName": "mysql-connectivity-service.occne-ndb",
  "udrServices": "All",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "sbiCorrelationInfoEnable": false,
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceed-44bb-95e0-f6a55a328b03",
  "vsaDefaultBillingDay": 1,

```



```

"subscriberActivityEnabled": false,
"addDefaultBillingDay": true,
"enableControlledShutdown" : true,
"keyType": "imsi",
"keyRange": "2013020001-2013020002",
"nfType": "UDR",
"nfFqdn": "udr001.oracle.com",
"suppressNotificationEnabled": true,
"configClientConnectTimeout": 1000,
"configClientReadTimeout": 1000,
"subscriberIdentifiers": {
    "msisdn": [],
    "imsi": [],
    "nai": [],
    "extid": []
}
}

```

The following REST API's are supported for auto create feature:

Note

Diameter requests for PUR and SNR are applicable for auto create subscribes.

Table 2-23 Supported APIs for Auto Create

Operation Type	Resource Name	REST URI
PUT	ProfileData	{apiRoot}/nudr-dr-prov/v2/profile-data/{ue-id}
PUT	AM,SM,UEPolicySet	{apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}
PUT	Subscriber/Quota/State/DynamicQuota	{apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa
PATCH	Subscriber	{apiRoot}/nudr-dr-prov/v2/policy-data/{ueId}/sm-data/vsa/subscriber

To Observe

See UDR [Metrics](#) section for information on metrics.

2.20 ETag (Entity Tag) support for UDR

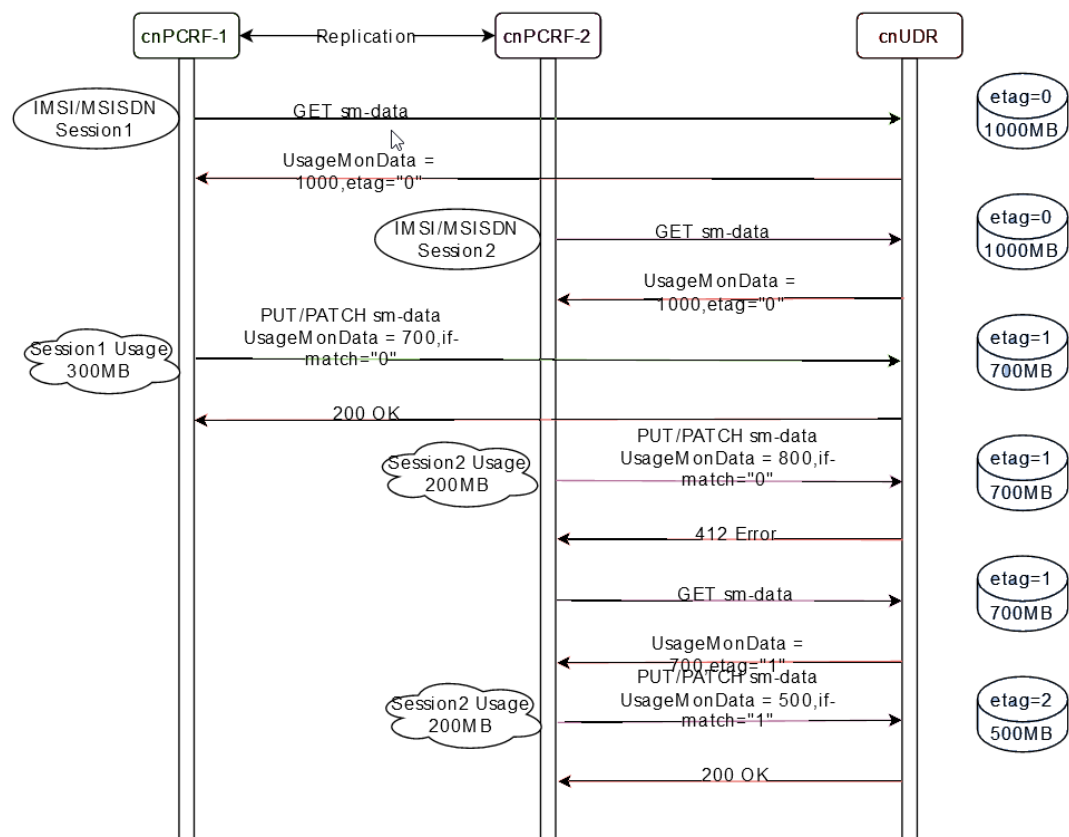
ETag (Entity Tag) support for UDR helps to make sure that the update is successful only when the consumer has the latest set of data, which is determined by the ETag value received at UDR. When two different Cloud Native Policy and Charging Rules Function (cnPCRF) are connected to UDR in a georedundant setup and both the cnPCRF are updating the

UsageMonData for the same subscriber simultaneously, there is possibility of data loss as the update can overwrite each other.

ETag is introduced in the HTTP header to resolve this data loss. When you send a GET sm-data request to update the UsageMonData, the ETag header with default value 0 is attached with the sm-data and added in the GET sm-data response. This ETag value is used in PUT or PATCH update request with if-match header to maintain concurrency control.

When the PUT or PATCH update request are received by database with if-match header, the database compares the header ETag value with the value attached with the sm-data. If both the values match, the request is processed and the ETag is updated with a different value, for example: 1, and this updated value is sent back in the PUT or PATCH response. If the ETag value does not match, the data update fails with error code 412 (etag out of sync). All provisioning and signaling operations to the sm-data will increment the ETag value.

Figure 2-15 ETag in HTTP response



In the call flow above, If both PCRF 1 and PCRF 2 sends a GET sm-data request to update the UsageMonData, both the cnPCRFs receive data with ETag value "0". When cnPCRF 1 send a PUT or PATCH request with "if-match" in the request header, the ETag value is validated by the database and data is updated with new Etag value "1". If cnPCRF 2 tries to update UsageMonData with older ETag value "0", UDR rejects the request with 412 error code (etag out of sync). The cnPCRF need to fetch the updated sm-data from UDR and retry the request with the updated ETag value to successfully update the data.

When the request is send with "If-None-Match" scenario to find if the latest data is available in UDR. If-None-Match is successful the UDR has the latest data and this data need to be send

to the cnPCRF. If-None-Match is failed the UDR has the older data and UDR send 304 response without any content in the response.

ETag in HTTP Response Scenarios

ETag for GET Response

Request:

```
GET /nudr-dr/v2/policy-data/ues/imsi-4500810000000018/sm-data HTTP/2
Host: ocudr-ingressgateway-sig
```

Response:

```
HTTP/2 200 OK
date: Wed, 11 Jan 2023 09:59:03 GMT
ETag: "0"
Content-Length: 546
Content-type: application/json
```

GET response when If-None-Match is success

Request:

```
GET /nudr-dr/v2/policy-data/ues/imsi-4500810000000018/sm-data HTTP/2
Host: ocudr-ingressgateway-sig
if-none-match: "1"
```

Response:

```
HTTP/2 200 OK
date: Wed, 11 Jan 2023 09:59:03 GMT
ETag: "0"
Content-Length: 546
Content-type: application/json
```

GET response when If-None-Match is failed

Request:

```
GET /nudr-dr/v2/policy-data/ues/imsi-4500810000000018/sm-data HTTP/2
Host: ocudr-ingressgateway-sig
if-none-match: "0"
```

Response:

```
HTTP/2 304
date: Wed, 11 Jan 2023 09:59:03 GMT
Content-type: application/json
```

ETag for PATCH operation

PATCH request when If-Match is successful

Request:

```
PATCH /nudr-dr/v2/policy-data/ues/imsi-450081000000018/sm-data HTTP/2
Host: ocudr-ingressgateway-sig
If-Match: "0"
Content-Length: 496
Content-type: application/json
```

Response:

```
HTTP/2 204 No Content
date: Wed, 11 Jan 2023 09:59:16 GMT
ETag: "1"
Content-Length: 0
```

PATCH request when If-Match is failed

Request:

```
PATCH /nudr-dr/v2/policy-data/ues/imsi-450081000000018/sm-data HTTP/2
Host: ocudr-ingressgateway-sig
If-Match: "2"
Content-Length: 496
Content-type: application/json
```

Response:

```
HTTP/2 412
date: Wed, 11 Jan 2023 09:59:44 GMT
Content-Length: 0
```

Etag Notifications

When usage monitoring data update is complete, UDR sends updated ETag notifications header to the consumer about the current state of the usage monitoring data. This enables the consumer to update the ETag values in the subsequent update operations.

Figure 2-16 Etag Notifications for Provisioning Requests

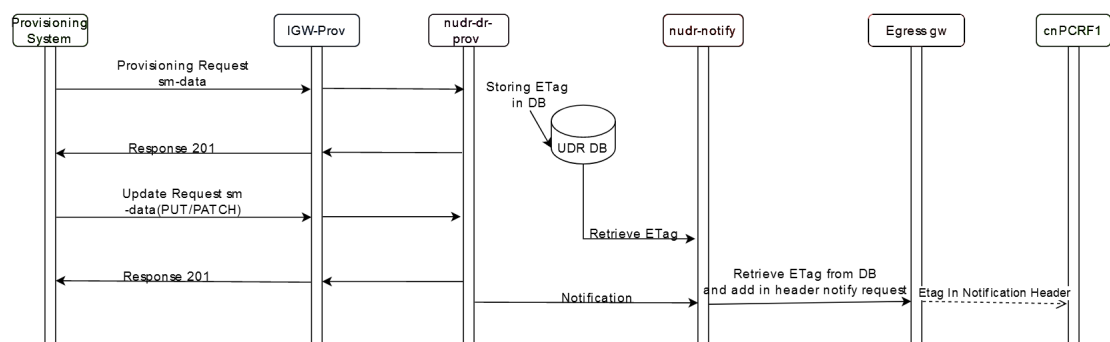
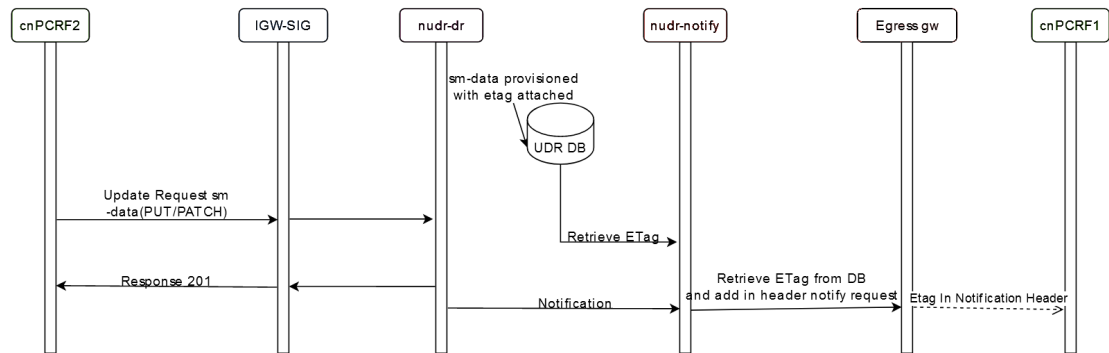


Figure 2-17 Notification for Signaling Requests

To Enable

You can enable or disable ETag feature by setting the `etagEnabled` parameter to true or false in the global section of the `ocudr-custom-values.yaml` file. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

global:

```
# Flag to enable/disable the addition of etag
etagEnabled: true
```

You can enable or disable the feature by setting the `etagEnabled` parameter to true or false in the [Global Configurations](#) in the CNC Console.

Configure using REST API:

You can perform the PUT operation on the `http://nudr-config-service:PORT/nudr-config/v1/udr.global.cfg/GLOBAL` to configure the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "dbServiceName": "mysql-connectivity-service.occne-ndb",
  "udrServices": "All",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "sbiCorrelationInfoEnable": false,
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "vsaDefaultBillingDay": 1,
  "subscriberActivityEnabled": false,
  "addDefaultBillingDay": true,
}
```

```

    "enableControlledShutdown" : true,
    "keyType": "imsi",
    "keyRange": "2013020001-2013020002",
    "nfType": "UDR",
    "nfFqdn": "udr001.oracle.com",
    "suppressNotificationEnabled": true,
    "configClientConnectTimeout": 1000,
    "configClientReadTimeout": 1000,
    "subscriberIdentifiers": {
        "msisdn": [],
        "imsi": [],
        "nai": [],
        "extid": []
    }
}

```

The following operations are supported for signaling using REST API:

- PATCH: `https://localhost:9443/nudr-dr/v2/policy-data/ues/msisdn-1111111111/sm-data`
- PUT and DELETE: `https://localhost:9443/nudr-dr/v2/policy-data/ues/msisdn-1111111111/sm-data/mk1`

The following operation is supported for provisioning using REST API:

- PUT: `http://localhost:8080/nudr-dr-prov/v2/policy-data/msisdn-461772271137732`

To Observe

The following metrics are introduced for ETag feature:

- `udr_etag_rest_request_total`
- `udr_etag_rest_response_total`
- `nudr_notif_etag_notifications_sent_total`

See UDR [Metrics](#) section for more information on metrics.

Logging

Logs are generated only when there is a mismatch of the ETag value.

Upgrade and Rollback Support

When UDR is upgraded from the previous release version with `etagEnabled` flag set to true the below scenarios can occur:

- The new data created will have ETag value generated and attached to the data and the subsequent update operations will use this ETag value.
- The existing data which does not have ETag value attached, will send the default ETag value 0 in GET response header and new a ETag value is generated and attached to the data in the next update operation.

When rollback is performed on UDR to the previous release version, if there is ETag value attached to the data, the GET response does not add the ETag value in the response header and ETag validation is not performed for subsequent update operations even if the "If-match" header is sent in the request.

2.21 Subscriber Export Tool

The subscriber export tool is part of the UDR tools and is used to export the subscriber database from UDR. The subscriber export tool is deployed in the same namespace of SLF and implemented as a stand alone tool. A new nudr-export-tool microservice is introduced and it provides the following features:

- Supports configuration on periodicity of export and export time
- Supports monitoring the export status of the subscriber data
- Supports transfer of exported files in .csv, .exml, or .txt format to the remote server
- Supports deletion of exported files from PVC (Persistent Volume Claim) when files are successfully exported to the remote server
- Supports on-demand export of the subscriber data
- Supports secure transfer of exported files to the remote server using Secure File Transfer Protocol (SFTP)

Scheduling Subscriber Export Tool

The subscriber export tool supports configuration of export files at periodic intervals. The export frequency given below are configurable.

- DAILY
- WEEKLY
- MONTHLY
- ONE TIME (used for ondemand export of subscriber data)

The start time of the export must be configured along with the configured frequency. You must use day-month-year-hours-minutes-seconds (dd-MM-YYYY hh:mm:ss) and 24 hours format for configuration. For more information, see [Customizing Subscriber Export Tool](#).

2.21.1 Supports Export of SLF and EIR Subscriber Database

This feature supports periodic exporting of SLF subscriber data into Comma Separated Value (CSV) file format and supports periodic exporting of EIR subscriber database containing all International Mobile Equipment Identity (IMEI) and associated International Mobile Subscriber Identity (IMSI), and Mobile Station Integrated Services Digital Network (MSISDN) using the subscriber export tool. The subscriber export tool is scheduled for periodic and on-demand runs and is deployed as a stand-alone tool. It also supports dynamic configuration of periodicity of export. For more information on subscriber export tool, see [Subscriber Export Tool](#).

Configure using CNC Console:

You can configure different parameters of subscriber export tool using CNC Console. For more information, see [Subscriber Export Tool Configurations](#) in CNC Console.

Configure using REST API:

You can perform the GET and PUT operations to configure the feature. For more information about REST based APIs, see the **Configuration APIs for UDR Microservices** section in the **Configuration API Specification** chapter of the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Observe

See [UDR Metrics](#) and [Alert Details](#) sections for more information on metrics and configurable parameters, respectively.

2.21.1.1 Exporting SLF Subscriber Database

The subscriber database is exported to the files in the below format. For keys not present, the value given is empty:

```
bash-4.4$ cat 2022-10-13-07-34-10-012118459_SLFsegment1.txt
accountId,imsi,nai,msisdn,extid,SLFGroupName,CHFGROUPID
10000000000000001001000002,,1001000002,,IMSGRP2,
10000000000000001001000004,,1001000004,,IMSGRP4,
```

The field delimiters and multiple value delimiters are configurable. For more information, see [Customizing Subscriber Export Tool](#). The format of the file name is as below:

```
timestamp_<ConfigurableFileName>.<configurable extension>
```

The maximum file size for each exported file is 1 GB. If the exported file exceeds the maximum limit, then the file will split into multiple files.

If the subscriber data is larger than the maximum limit then the data is split into separate files. The files are given naming extension to indicate the order of files. For example, if the subscriber data is split into 3 files the files created are:

- 2022-09-21-16-50-01-578894500_SLFTestSegment-1.txt → First set of subscribers
- 2022-09-21-16-50-01-578894500_SLFTestSegment-2.txt → Second set of subscribers
- 2022-09-21-16-50-01-578894500_SLFTestSegment.txt → Last set of subscribers

2.21.1.2 Exporting EIR Subscriber Database

The subscriber database is exported to the files in the below format:

```
bash-4.4$ cat 2022-11-12-21-10-00-211378929_EIRTestSegment.txt
IMEI,SVN,ALLOW,GRAY,BLOCK,IMSI1,IMSI2,IMSI3,IMSI4,IMSI5,MSISDN1,MSISDN2,MSISDN
3,MSISDN4,MSISDN5
11111111111111,0,y,y,y,1111111121,1111111131,0,0,0,1111111113,1111111112,0,0
,0
```

 Note

Multiple value delimiters are not applicable for EIR export, it is applicable only for SLF export.

The format of the file name is as below:

```
timestamp_<ConfigurableFileName>.<configurable extension>
```


The count of IMSIs and MSISDNs are configurable. For example, if five IMSIs and MSISDNs are configured but not provisioned for that IMEI, then 0 is used.

The maximum file size for each exported file is 1 GB. If the exported file exceeds the maximum limit, then the file will split into multiple files.

If the subscriber data is larger than the maximum limit then the data is split into separate files. The files are given naming extension to indicate the order of files. For example, if the subscriber data is split into 2 files the files created are:

- 2022-11-12-21-10-00-211378929_EIRTestSegment.txt-1.txt → First set of subscribers
- 2022-11-12-21-10-00-211378929_EIRTestSegment.txt-2.txt → Second set of subscribers

2.21.2 Support for EXML Format

This feature is introduced to export the subscriber data in EXML format, which is compatible with 4G OCUDR export format. It supports the export of 4G policy data (VSA and umData/umDataLimits) in EXML format from 5G UDR to 4G OCUDR using the subscriber export tool and hence, supports rollback of exported data. The subscriber export tool supports the following functionalities to export the subscriber data in the .exml format:

- Supports configuring the export file to .exml format
- Supports configuration on periodicity of export and export time
- Supports monitoring the export status of the subscriber data
- Supports transfer of exported files in .exml format to the remote server using sftpEnabled flag.
- Supports the deletion of exported files from PVC (Persistent Volume Claim) when files are successfully exported to the remote server using deletePVC flag.

For more information about subscriber export tool, see [Subscriber Export Tool](#).

Note

When EXML file is generated using nudr-export-tool micro service, you must convert the EXML file to ixml file using ixml converter tool from 4G UDR. Once the file is converted to ixml format, use nudr-bulk-import micro service for import.

The subscriber export tool will make following subscriber keys in upper case in the EXML file:

- MSISDN
- IMSI
- NAI
- EXTID
- ACCOUNTID

Export of 4G Policy Data

The subscriber export tool is enhanced to export the 4G policy subscriber data to .exml format in the following two methods:

- Export the total range of 4G policy subscriber data from the 5G UDR database
- Export of selected range of 4G policy subscriber data from the 5G UDR database

Export the Total Range of Subscriber Data

To export the total range of 4G policy subscriber data, you must set the value of the following parameters in the EXPORT_TOOL_CFG payload as follows:

Table 2-24 Export Tool Parameter Values

Parameter	Values
exportMode	EXML_EXPORT
fileExtension	.exml
subscriberRangeEnabled	false

Configure using REST API:

You can perform a GET or PUT operation on the `http://<nudr-config-url:port>/nudr-config/v1/udr.exporttool.cfg/EXPORT_TOOL_CFG` to export the total range of subscriber data. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "remotePath": "/home/cloud-user/ocudr/",
  "fixedDumpFileNameSuffix": "SLFTestSegment1",
  "msisdnCount": 6,
  "pvcPath": "/home/udruser/export1",
  "fieldDelimiter": ",",
  "multiValuedDelimiter": "|",
  "subscriberExportRange": {
    "keyType": "msisdn",
    "subscriberRanges": [
      "1111111130-1111111140",
      "1111111150-1111111160"
    ]
  },
  "subscriberRangeEnabled": false,
  "fileExtension": ".txt",
  "deleteFromPVC": true,
  "startTime": "14/09/2022 17:32:10",
  "imsiCount": 5,
  "newFilePerRangeEnabled": false,
  "frequencyOfPeriodicDump": "DAILY",
  "maxNumberOfDumps": 5,
  "exportFileExpiryTime": 7d,
  "policyDataResources": "sm-data, am-data, ue-policy-set"
}
```

Note

For EXML mode, multiValuedDelimiter and fieldDelimiter parameters are not applicable.

Range Based Export of Subscriber Data

Range based support helps to migrate the 4G policy subscriber data to 5G UDR in a phase manner. This support is available for Mobile Station Integrated Services Digital Network (MSISDN) or International Mobile Subscriber Identity (IMSI) key types. You can configure a single key (MSISDN or IMSI) with a maximum range support of five ranges per key.

You can perform the range based subscriber data export using any one of the following methods:

- Export of 4G policy subscriber data to a single exml file.
- Export of 4G policy subscriber data to multiple exml files for each range.

Export of 4G Policy Subscriber Data to a Single EXML Files.

To enable the export of subscriber data to a single exml files, you must set the value of the following parameters in the EXPORT_TOOL_CFG payload as follows:

Table 2-25 Export Tool Parameter Values

Parameter	Values
subscriberRangeEnabled	true
exportMode	EXML_EXPORT
fileExtension	.exml
keyType	MSISDN or IMSI
subscriberRanges	["1111111130-1111111140", "1111111150-1111111160"]

Configure using REST API:

You can perform a GET or PUT operation on the `http://<nudr-config-url:port>/nudr-config/v1/udr.exporttool.cfg/EXPORT_TOOL_CFG` to export the subscriber data to a single exml file. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "remotePath": "/home/cloud-user/ocudr/",
  "fixedDumpFileNameSuffix": "SLFTestSegment1",
  "msisdnCount": 6,
  "pvcPath": "/home/udruser/export1",
  "fieldDelimiter": ",",
  "multiValuedelimiter": "|",
  "subscriberExportRange": {
    "keyType": "msisdn",
    "subscriberRanges": [
      "1111111130-1111111140",
      "1111111150-1111111160"
    ]
  },
  "subscriberRangeEnabled": false,
```

```

    "fileExtension": ".txt",
    "deleteFromPVC": true,
    "startTime": "14/09/2022 17:32:10",
    "imsiCount": 5,
    "newFilePerRangeEnabled": false,
    "frequencyOfPeriodicDump": "DAILY",
    "maxNumberOfDumps": 5,
    "exportFileExpiryTime": 7d,
    "policyDataResources": "sm-data, am-data, ue-policy-set"
  }

```

Export of 4G Policy Subscriber Data to Multiple Exml File

To enable the export of subscriber data to multiple exml file, you must set the value of the following parameters in the EXPORT_TOOL_CFG payload as follows:

Table 2-26 Export Tool Parameter Values

Parameter	Values
subscriberRangeEnabled	true
newFilePerRangeEnabled	true
exportMode	EXML_EXPORT
fileExtension	.exml
keyType	MSISDN or IMSI
subscriberRanges	["1111111130-1111111140", "1111111150-1111111160"]

Note

To support multiple exml files, you must enable both subscriberRangeEnabled and newFilePerRangeEnabled parameters to true.

Configure using REST API:

You can perform a GET or PUT operation on the `http://<nudr-config-url:port>/nudr-config/v1/udr.exporttool.cfg/EXPORT_TOOL_CFG` to export the subscriber data to multiple exml files. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```

{
  "remotePath": "/home/cloud-user/ocudr/",
  "fixedDumpFileNameSuffix": "SLFTestSegment",
  "msisdnCount": 5,
  "pvcPath": "/home/udruser/export",
  "fieldDelimiter": ",",
  "multiValueddelimiter": "|",
  "subscriberExportRange": {
    "keyType": "msisdn",

```

```

        "subscriberRanges": [
            "1111111130-1111111140",
            "1111111150-1111111160"
        ],
        "subscriberRangeEnabled": false,
        "fileExtension": ".txt",
        "deleteFromPVC": true,
        "startTime": "14/09/2022 17:32:10",
        "imsiCount": 5,
        "newFilePerRangeEnabled": false,
        "frequencyOfPeriodicDump": "DAILY",
        "maxNumberOfDumps": 5,
        "exportFileExpiryTime": 7d,
        "policyDataResources": "sm-data, am-data, ue-policy-set"
    }
}

```

Configure using CNC Console:

You can configure different EXMLsupport parameters in the subscriber export tool using CNC Console. For more information, see [Subscriber Export Tool Configurations](#) in CNC Console.

For more information on configuring subscriber export tool, see [Customizing Subscriber Export Tool](#).

2.21.3 Support for Export of Policy Data in Comma Separated Value (CSV) Format

This feature enables the Subscriber Export Tool to export the 5G and 4G subscriber policy data, which includes the profile data and policy data (am-data, sm-data, and ue-policy-set) in CSV file format from cnUDR. The converted subscriber data in CSV file format is used by the Subscriber Bulk Import Tool to import the subscriber data along with its policy data on another instance of cnUDR.

To support the this feature, the exportMode parameter is updated to include POLICY_DATA_EXPORT mode, which supports the following functionalities:

- On-demand export of the subscriber data.
- Periodic export of the subscriber data. The periodicity are DAILY, WEEKLY, and MONTHLY.
- Export of the subscriber data based on the range. This support is available for Mobile Station Integrated Services Digital Network (MSISDN) and International Mobile Subscriber Identity (IMSI) key types.

The following functionalities are supported to export the subscriber data in the .csv format using the existing configurations on the Subscriber Export Tool:

- Supports configuration on periodicity of export and export time.
- Supports generating the export file to .csv format.
- Supports monitoring the export status of the subscriber data.
- Supports transfer of exported files in .csv format to the remote server using sftpEnabled flag.

- Supports the deletion of exported files from PVC (Persistent Volume Claim) when files are successfully exported to the remote server using deletePVC flag.

The following functionalities are supported for error handling:

- Export with the available data that is fetched.
- Timeout and retries for failed read database operations.
- Partial export of subscriber data is supported.

Sample Format for the Exported 5G Policy Subscriber Data

The following sample payload includes policy data (am-data, sm-data, and ue-policy-set) and profile data under VendorSpecific-000111 parameter and under sm-data JSON object.

```
keyType,keyValue,operationType,nfType,jsonPayloadimsi,1111111111,MODIFY,PCF,"{
  "am-data": { "praInfos": { "p1": { "praId": "p1", "trackingAreaList":
    [{ "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId":
    { "mcc": "977", "mnc": "33" }, "tac": "5CB7" } ] }, "ecgiList": [{ "plmnId":
    { "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, { "plmnId":
    { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" } ] }, "ncgiList":
    [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" },
    { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" } ] },
    "globalRanNodeIdList": [{
      "plmnId": {
        "mcc": "965", "mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" }, {
      "plmnId": {
        "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25,
        "gNbValue": "1A8F1D" } } ] } }, "subscCats": [ "cat1", "cat2" ] }, "sm-data":
    { "umData": { "mk1": { "scopes": { "11-abc123": { "dnn": [ "dnn1" ],
    "snssai": { "sd": "abc123", "sst": 11 } } } }, "limitId": "mk1", "name":
    "Weekend", "umLevel": "SERVICE_LEVEL", "resetTime":
    "2018-01-02T08:17:14.090Z",
    "allowedUsage": {
      "duration": 9000,
      "totalVolume": 8888,
      "uplinkVolume": 6666,
      "downlinkVolume": 7777
    }, "customAllowedUsage": { "duration":
    9000, "totalVolume":
    8888, "downlinkVolume":
    6666, "uplinkVolume":
    7777 }, "serviceSpecific": 7777, "type":
    "", "grantedUsage": { "totalVolume": 8888, "downlinkVolume": 6666,
    "uplinkVolume": 7777, "duration": 3000 }, "grantedServiceSpecific":
    "", "quotaState": "",
    "refInstanceId":
    "" } }, "umDataLimits": { "mk1": { "scopes": { "11-abc123":
    { "dnn": [ "dnn1" ], "snssai": { "sd": "abc123", "sst": 11 } } } }, "endDate":
    "2018-11-05T08:17:14.090Z", "limitId": "mk1", "umLevel": "SESSION_LEVEL",
    "startDate": "2018-09-05T08:17:14.090Z",
    "usageLimit": {
      "duration": 6000,
      "totalVolume": 9000,
      "uplinkVolume": 5000,
      "downlinkVolume": 4000
    }, "resetPeriod": {
```

```
"period":  
    "MONTHLY" },  
    "Type": "", "name": "DynamicAggregateLimit", "priority": "",  
    "initialServiceSpecific": 4000, "activationdatetime":  
    "2021-09-05T08:17:14.090Z",  
    "validityDuration":  
        300, "InterimReportingInterval": "" } }, "smPolicySnssaiData":  
{ "11-abc123": { "snssai": { "sd": "abc123", "sst": 11 }, "smPolicyDnnData":  
{ "dnn1": { "dnn": "dnn1", "bdtRefIds": { "xyz": "bdtRefIds", "abc": "xyz" },  
"gbrDL": "7788 Kbps", "gbrUL": "5566 Kbps", "online": true, "chfInfo":  
{ "primaryChfAddress": "1.1.1.1", "secondaryChfAddress": "2.2.2.2" },  
"offline": true, "praInfos": { "p1": { "praId": "p1", "trackingAreaList":  
[{"plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, {"plmnId":  
{ "mcc": "977", "mnc": "33" }, "tac": "5CB7" } ] }, "ecgiList": [{"plmnId":  
{ "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, {"plmnId":  
{ "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" } ], "ncgiList":  
[{"plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" },  
{ "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" } ] },  
"globalRanNodeIdList": [{  
    "plmnId": {  
        "mcc": "965", "mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" }, {  
    "plmnId": {  
        "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25,  
"gNbValue": "1A8F1D" } } ] } }, "ipv4Index": 0, "ipv6Index": 0, "subscCats":  
[ "cat1", "cat2" ], "adcSupport": true, "mpsPriority": true,  
"allowedServices": [ "ser1", "ser2" ], "mpsPriorityLevel": 2,  
"imsSignallingPrio": true,  
    "refUmDataLimitIds": {  
        "mk1": { "monkey": [ "monkey1" ], "limitId": "mk1" } },  
"subscSpendingLimits": true } } } },  
    "VendorSpecific-000111": { "subscriber": {  
        "IMSI": ["7070707070"], "ACCOUNTID": ["8080808080808080"],  
"MSISDN": ["2012012012", "1021021021"] } } }, "ue-policy-set": { "praInfos":  
{ "p1": { "praId": "p1", "trackingAreaList": [{"plmnId": { "mcc": "976",  
"mnc": "32" }, "tac": "5CB6" }, {"plmnId": { "mcc": "977", "mnc": "33" },  
"tac": "5CB7" } ], "ecgiList": [{"plmnId": { "mcc": "976", "mnc": "32" },  
"eutraCellId": "92FFdBE" }, {"plmnId": { "mcc": "977", "mnc": "33" },  
"eutraCellId": "8F868C4" } ], "ncgiList": [{"plmnId": { "mcc": "976", "mnc":  
"32" }, "nrCellId": "b2fB6fE9D" }, {"plmnId": { "mcc": "977", "mnc": "33" },  
"nrCellId": "5d1B4127b" } ] }, "globalRanNodeIdList": [{"plmnId": { "mcc":  
"965", "mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" }, {  
    "plmnId": {  
        "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25,  
"gNbValue": "1A8F1D" } } ] } }, "subscCats": [ "cat1", "cat2" ],  
"uePolicySections": { "33": { "upsi": "33", "uePolicySectionInfo": "MQ==" },  
"111": { "upsi": "111", "uePolicySectionInfo": "Mg0K" } }, "upsis": [ "111",  
"33" ], "allowedRouteSelDescs": {  
    "965-235": {  
        "servingPlmn": {  
            "mcc": "965", "mnc": "235" }, "snssaiRouteSelDescs": [{  
                "snssai": {  
                    "sd": "abc123", "sst": 11 }, "dnnRouteSelDescs": [{"dnn": "dnn1",  
"sscModes": [ "SSC_MODE_1", "SSC_MODE_2" ], "pduSessTypes":  
[ "IPV4" ] } ] } } }, "andspInd": true, "pei": "imei-355375345274757",  
"osIds": [ "bceafb30-31bd-4b8d-b3c2-ed79d029531a",  
"cc9d9f97-0127-497c-9bca-79e8e9080ea8" ] } }
```

Sample Format for the Exported 4G Policy Subscriber Data

Note

The 4G policy data is converged to the 5G data model.

The following sample payload includes policy data (am-data, sm-data, and ue-policy-set) and 4G subscriber data, and state information data under VendorSpecific-000111 parameter under sm-data JSON object.

```
keyType,keyValue,operationType,nfType,jsonPayloadimsi,1111111111,MODIFY,PCF,"{
  "am-data": { "praInfos": { "p1": { "praId": "p1", "trackingAreaList":
    [{ "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId":
    { "mcc": "977", "mnc": "33" }, "tac": "5CB7" } ], "ecgiList": [{ "plmnId":
    { "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, { "plmnId":
    { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" } ], "ncgiList":
    [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" },
    { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" } ],
    "globalRanNodeIdList": [{
      "plmnId": {
        "mcc": "965", "mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" }, {
      "plmnId": {
        "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25,
        "gNbValue": "1A8F1D" } } ] } }, "subscCats": [ "cat1", "cat2" ] }, "sm-data":
    { "umData": { "mk1": { "scopes": { "11-abc123": { "dnn": [ "dnn1" ],
    "snssai": { "sd": "abc123", "sst": 11 } } }, "limitId": "mk1", "name":
    "Weekend", "umLevel": "SERVICE_LEVEL", "resetTime":
    "2018-01-02T08:17:14.090Z",
    "allowedUsage": {
      "duration": 9000,
      "totalVolume": 8888,
      "uplinkVolume": 6666,
      "downlinkVolume": 7777
    }, "customAllowedUsage": { "duration":
    9000, "totalVolume":
    8888, "downlinkVolume":
    6666, "uplinkVolume":
    7777 }, "serviceSpecific": 7777, "type":
    "", "grantedUsage": { "totalVolume": 8888, "downlinkVolume": 6666,
    "uplinkVolume": 7777, "duration": 3000 }, "grantedServiceSpecific":
    "", "quotaState": "",
    "refInstanceId":
    "" } }, "umDataLimits": { "mk1": { "scopes": { "11-abc123":
    { "dnn": [ "dnn1" ], "snssai": { "sd": "abc123", "sst": 11 } } }, "endDate":
    "2018-11-05T08:17:14.090Z", "limitId": "mk1", "umLevel": "SESSION_LEVEL",
    "startDate": "2018-09-05T08:17:14.090Z",
    "usageLimit": {
      "duration": 6000,
      "totalVolume": 9000,
      "uplinkVolume": 5000,
      "downlinkVolume": 4000
    }, "resetPeriod": {
      "period":
```



```

    "MONTHLY" },
    "Type": "", "name": "DynamicAggregateLimit", "priority": "",
    "initialServiceSpecific": 4000, "activationdatetime":
    "2021-09-05T08:17:14.090Z",
    "validityDuration":
    300, "InterimReportingInterval": "" } }, "smPolicySnssaiData":
{ "11-abc123": { "snssai": { "sd": "abc123", "sst": 11 }, "smPolicyDnnData":
{ "dnn1": { "dnn": "dnn1", "bdtRefIds": { "xyz": "bdtRefIds", "abc": "xyz" },
"gbrDL": "7788 Kbps", "gbrUL": "5566 Kbps", "online": true, "chfInfo":
{ "primaryChfAddress": "1.1.1.1", "secondaryChfAddress": "2.2.2.2" },
"offline": true, "praInfos": { "p1": { "praId": "p1", "trackingAreaList":
[ { "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId":
{ "mcc": "977", "mnc": "33" }, "tac": "5CB7" } ] }, "ecgiList": [ { "plmnId":
{ "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, { "plmnId":
{ "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" } ], "ncgiList":
[ { "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" },
{ "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" } ] },
"globalRanNodeIdList": [ {
    "plmnId": {
        "mcc": "965", "mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" }, {
        "plmnId": {
            "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25,
            "gNbValue": "1A8F1D" } } ] } }, "ipv4Index": 0, "ipv6Index": 0, "subscCats":
[ "cat1", "cat2" ], "adcSupport": true, "mpsPriority": true,
"allowedServices": [ "ser1", "ser2" ], "mpsPriorityLevel": 2,
"imsSignallingPrio": true,
    "refUmDataLimitIds": {
        "mk1": { "monkey": [ "monkey1" ], "limitId": "mk1" } },
    "subscSpendingLimits": true } } } },
    "VendorSpecific-000111": { "subscriber": {
        "IMSI": ["302370123456789"], "ACCOUNTID": ["178322212122"],
        "MSISDN": ["15145551234", "15141234567"], "BillingDay": 6, "Entitlement":
        [ "DayPass", "DayPassPlus" ] }, "state": { "property": [ { "name": "mcc",
        "value": "123" }, { "name": "expire", "value": "201402-09T11:20:32" } ] },
        "version": 1 } } },
    "ue-policy-set": { "praInfos": { "p1": { "praId": "p1",
    "trackingAreaList": [ { "plmnId": { "mcc": "976", "mnc": "32" }, "tac":
    "5CB6" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "tac": "5CB7" } ] },
    "ecgiList": [ { "plmnId": { "mcc": "976", "mnc": "32" }, "eutraCellId":
    "92FFdBE" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "eutraCellId":
    "8F868C4" } ], "ncgiList": [ { "plmnId": { "mcc": "976", "mnc": "32" },
    "nrCellId": "b2fB6fE9D" }, { "plmnId": { "mcc": "977", "mnc": "33" },
    "nrCellId": "5d1B4127b" } ] }, "globalRanNodeIdList": [ { "plmnId": { "mcc":
    "965", "mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" }, {
        "plmnId": {
            "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25,
            "gNbValue": "1A8F1D" } } ] } }, "subscCats": [ "cat1", "cat2" ],
    "uePolicySections": { "33": { "upsi": "33", "uePolicySectionInfo": "MQ==" },
    "111": { "upsi": "111", "uePolicySectionInfo": "Mg0K" } }, "upsis": [ "111",
    "33" ], "allowedRouteSelDescs": {
        "965-235": {
            "servingPlmn": {
                "mcc": "965", "mnc": "235" }, "snssaiRouteSelDescs": [ {
                    "snssai": {
                        "sd": "abc123", "sst": 11 }, "dnnRouteSelDescs": [ { "dnn": "dnn1",
                        "sscModes": [ "SSC_MODE_1", "SSC_MODE_2" ], "pduSessTypes":

```

```
[ "IPV4" ] } } } } }, "andspInd": true, "pei": "imei-355375345274757",
"osIds": [ "bceafb30-31bd-4b8d-b3c2-ed79d029531a",
"cc9d9f97-0127-497c-9bca-79e8e9080ea8" ] } } }
```

Managing Support for Policy Data Export in CSV Format

Configure

The `exportMode` parameter is updated to include `POLICY_DATA_EXPORT` mode. The following modes are allowed for the `exportMode` parameter:

- `POLICY_DATA_EXPORT`
- `CSV_EXPORT`
- `PDBI_EXPORT`
- `EXML_EXPORT`

For more information on the configurations, see [Customizing Subscriber Export Tool](#).

A new configuration `policyDataResources` and `exportFileExpiryTime` is added in the `EXPORT_TOOL_CFG` payload to indicate the policy data information that is exported. By default `am-data`, `sm-data`, and `ue-policy-set` is read.

Table 2-27 Configuring Policy Data Resources

Configuration Name	Description	Value
<code>policyDataResources</code>	It is the type of data that needs to be exported for the policy data. By default, profile data is included. Data types must be added in CSV format.	Default value: <code>sm-data, am-data, ue-policy-set</code> Allowed value: Comma separated resource names. One of the following values is mandatory: • <code>am-data</code> • <code>am-data, ue-policy-set</code> • <code>sm-data, am-data</code>
<code>exportFileExpiryTime</code>	The exported file expiry time, after which the file is deleted.	Default value: 7d

Export the Total Range of Subscriber Data

To export the total range of subscriber data, you must set the value of the following parameters in the `EXPORT_TOOL_CFG` payload as follows:

Table 2-28 Subscriber Export Tool Parameter Values

Parameter	Values
<code>exportMode</code>	<code>POLICY_DATA_EXPORT</code>
<code>fileExtension</code>	<code>.csv</code>
<code>subscriberRangeEnabled</code>	<code>false</code>
<code>fieldDelimiter</code>	<code>,</code>

Configure using REST API

You can perform a GET or PUT operation on the `http://<nudr-config-url:port>/nudr-config/v1/udr.exporttool.cfg/EXPORT_TOOL_CFG` to export the total range of subscriber data. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "remotePath": "/home/cloud-user/ocudr/",
  "fixedDumpFileNameSuffix": "SLFTestSegment1",
  "msisdnCount": 6,
  "pvcPath": "/home/udruser/export1",
  "fieldDelimiter": ",",
  "multiValueddelimiter": "|",
  "subscriberExportRange": {
    "keyType": "msisdn",
    "subscriberRanges": [
      "1111111130-1111111140",
      "1111111150-1111111160"
    ]
  },
  "subscriberRangeEnabled": false,
  "fileExtension": ".txt",
  "deleteFromPVC": true,
  "startTime": "14/09/2022 17:32:10",
  "imsiCount": 5,
  "newFilePerRangeEnabled": false,
  "frequencyOfPeriodicDump": "DAILY",
  "maxNumberOfDumps": 5,
  "exportFileExpiryTime": 7d,
  "policyDataResources": "sm-data, am-data, ue-policy-set"
}
```

Range-Based Export of Subscriber Data

Range-based support helps to migrate the 4G policy subscriber data to 5G UDR in a phased manner. This support is available for Mobile Station Integrated Services Digital Network (MSISDN) or International Mobile Subscriber Identity (IMSI) key types. You can configure a single key (MSISDN or IMSI) with a maximum range support of five ranges per key.

You can perform the range-based subscriber data export using any one of the following methods:

- Export of 4G policy subscriber data to a single CSV file.
- Export of 4G policy subscriber data to multiple CSV files for each range.

Export of Subscriber Data to a Single CSV Files.

To enable the export of subscriber data to a single CSV file, you must set the value of the following parameters in the `EXPORT_TOOL_CFG` payload as follows:

Table 2-29 Export Tool Parameter Values

Parameter	Values
subscriberRangeEnabled	true
exportMode	POLICY_DATA_EXPORT
fileExtension	.csv

Table 2-29 (Cont.) Export Tool Parameter Values

Parameter	Values
fieldDelimiter	,
keyType	MSISDN or IMSI
subscriberRanges	["1111111130-1111111140", "1111111150-1111111160"]

Configure using REST API:

You can perform a GET or PUT operation on the `http://<nudr-config-url:port>/nudr-config/v1/udr.exporttool.cfg/EXPORT_TOOL_CFG` to export the subscriber data to a single csv file. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "remotePath": "/home/cloud-user/ocudr/",
  "fixedDumpFileNameSuffix": "SLFTestSegment1",
  "msisdnCount": 6,
  "pvcPath": "/home/udruser/export1",
  "fieldDelimiter": ",",
  "multiValuedDelimiter": "|",
  "subscriberExportRange": {
    "keyType": "msisdn",
    "subscriberRanges": [
      "1111111130-1111111140",
      "1111111150-1111111160"
    ]
  },
  "subscriberRangeEnabled": false,
  "fileExtension": ".txt",
  "deleteFromPVC": true,
  "startTime": "14/09/2022 17:32:10",
  "imsiCount": 5,
  "newFilePerRangeEnabled": false,
  "frequencyOfPeriodicDump": "DAILY",
  "maxNumberOfDumps": 5,
  "exportFileExpiryTime": 7d,
  "policyDataResources": "sm-data, am-data, ue-policy-set"
}
```

Export of Subscriber Data to Multiple CSV File

To enable the export of subscriber data to multiple CSV file, you must set the value of the following parameters in the `EXPORT_TOOL_CFG` payload as follows:

Table 2-30 Export Tool Parameter Values

Parameter	Values
subscriberRangeEnabled	true
newFilePerRangeEnabled	true
exportMode	POLICY_DATA_EXPORT
fileExtension	.csv
fieldDelimiter	,
keyType	MSISDN or IMSI
subscriberRanges	["1111111130-1111111140", "1111111150-1111111160"]

Note

To support multiple .csv files, you must enable both `subscriberRangeEnabled` and `newFilePerRangeEnabled` parameters to true.

To support multiple .csv files, you must enable both `subscriberRangeEnabled` and `newFilePerRangeEnabled` parameters to true. If a particular range of the exported file size reaches the **fileSize** limit, then the export file will roll over to a new file. There can be multiple export files for a single export range.

Configure using REST API:

You can perform a GET or PUT operation on the `http://<nudr-config-url:port>/nudr-config/v1/udr.exporttool.cfg/EXPORT_TOOL_CFG` to export the subscriber data to multiple CSV files. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "remotePath": "/home/cloud-user/ocudr/",
  "fixedDumpFileNameSuffix": "SLFTestSegment1",
  "msisdnCount": 6,
  "pvcPath": "/home/udruser/export1",
  "fieldDelimiter": ",",
  "multiValuedelimiter": "|",
  "subscriberExportRange": {
    "keyType": "msisdn",
    "subscriberRanges": [
      "1111111130-1111111140",
      "1111111150-1111111160"
    ]
  },
  "subscriberRangeEnabled": false,
  "fileExtension": ".txt",
  "deleteFromPVC": true,
  "startTime": "14/09/2022 17:32:10",
}
```

```

    "imsiCount": 5,
    "newFilePerRangeEnabled": false,
    "frequencyOfPeriodicDump": "DAILY",
    "maxNumberOfDumps": 5,
    "exportFileExpiryTime": 7d,
    "policyDataResources": "sm-data, am-data, ue-policy-set"
  }

```

Observe

There are no metrics, KPIs, or alerts available for this feature.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.21.4 Secure File Transfer Support for Subscriber Export Tool

Subscriber export tool support the capability to securely transfer the exported data to a remote server using Secure File Transfer Protocol (SFTP).

To Enable using REST API:

You can perform the PUT operation on the `http://10.75.215.124:32523/nudr-config/v1/udr.exporttool.cfg/EXPORT_TOOL_CFG` to configure the subscriber export tool and enable SFTP. For more information, see [Customizing Subscriber Export Tool](#) and *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```

{
  "fixedDumpFileNamePrefix": "SLFTTestSegment",
  "remoteUserName": "cloud-user",
  "exportMode": "PDBI_EXPORT",
  "remotePath": "/home/cloud-user/jaipal/",
  "pollingInterval": "* /60 * * * * *",
  "msisdNCount": 5,
  "pvcPath": "/home/udruser/export",
  "fieldDelimiter": ",",
  "serverPort": "22",
  "multiValuedDelimiter": "|",
  "serverHost": "10.75.229.18",
  "fileExtension": ".txt",
  "deleteFromPVC": true,
  "sftpExportEnabled": true,
  "startTime": "14/09/2022 17:32:10",
  "imsiCount": 5,
  "frequencyOfPeriodicDump": "DAILY",

```

```
"maxNumberOfDumps": 5
}
```

To Enable using CNC Console or Helm Configuration:

You can enable or disable secure file transfer support using CNC Console or Helm Configuration.

- You can enable or disable the SFTP by setting the `sftpDetails.sftpExportEnabled` parameter to true or false in custom-value.yaml file of the subscriber export tool. For more information on the parameters, see [Customizing Subscriber Export Tool](#).

```
sftpDetails:
  host: 10.75.229.18
  port: 22
  username: cloud-user
  path: /home/cloud-user/jaipal/
  sftpExportEnabled: true
  deleteFromPvc: true
  pollingInterval: "*/60 * * * * *"
  secrets:
    privatekey:
      name: ocudr-ssh-private-key
    publickey:
      name: ocudr-ssh-public-key
  fileSizeCheck: false
  transferProtocol: sftp
```

① Note

- `transferProtocol`: Transfers the files to remote location. The default protocol is SFTP. The two types of protocols supported are SFTP and SCP.
- `fileSizeCheck`: Checks the file size of the remote server once the file is successfully transferred to remote location. This confirms that the full log file is transferred to the remote location. This is an optional flag. If the flag to set to true, application will fetch the file size from the same remote location path.

- You can disable or enable the feature by setting the `RemoteTransferEnabled` parameter to true or false in the [Subscriber Export Tool Configurations](#) in the CNC Console.

① Note

Perform the below procedure only if SFTP is enabled.

When the keys are generated by the operator for the remote host, you must run the below command to create a kubernetes secrets for both keys.

```
kubectl create secret generic ocudr-ssh-private-key --from-file=id_rsa=/home/
cloud-user/ocudr/secrets/id_rsa -n <namespace>
kubectl create secret generic ocudr-ssh-public-key --from-file=id_rsa.pub=/
home/cloud-user/ocudr/secrets/id_rsa.pub -n <namespace>
```

2.21.5 Installing Subscriber Export Tool

You can install the Subscriber Export Tool using helm. While installing the export tool, the "exporttoolpersistentclaim" PVC for bulk import container is created.

Before installing the Subscriber Export Tool using helm charts, you are required to configure its parameters in the nudr-export-tool-custom-values.yaml. For information about the Subscriber Export Tool configurable parameters, see [Customizing Subscriber Export Tool](#).

Fresh Installation of the Subscriber Export Tool

Perform the following steps to install the Subscriber Export Tool as a separate deployment with different a release name than ocudr:

1. Extract the Subscriber Export Tool helm charts from the software or CSAR package and run the following command:

```
helm install <release name for export tool> -n <udr/slf namespace> nudr-export-tool/ -f nudr-export-tool-custom-values.yaml
```

2. Example of the Subscriber Export Tool pod deployed in UDR/SLF namespace is given below:

```
[cloud-user@cne-22-3-bastion-1 ~]$ kubectl get pods -n udr
NAME                                READY   STATUS    RESTARTS   AGE
export-nudr-export-tool-0           1/1     Running   0           3h33m
http-go-prov-859d479b8-mcxk9        1/1     Running   0           9d
ocudr-egressgateway-64dc4bf9d8-6pd44 1/1     Running   0           6h28m
ocudr-ingressgateway-prov-7f678497bc-cglgq 1/1     Running   0           6h28m
ocudr-ingressgateway-sig-7b64fcd7f5-58qgm 1/1     Running   0           6h28m
ocudr-nudr-config-6c7f944d57-n7lpt 1/1     Running   0           6h28m
ocudr-nudr-config-server-bf777b9cf-ncc2s 1/1     Running   0           6h28m
ocudr-nudr-dr-provservice-64b9f5f46c-mdhsb 1/1     Running   0           6h28m
ocudr-nudr-drservice-7bc4cc894b-hddcv 1/1     Running   0           6h28m
```

Note

Upgrading of the Subscriber Export Tool is not supported. Only clean installation of the Subscriber Export Tool is supported, all older configurations will be lost during clean installation.

For scheduling the subscriber data export, see **Scheduling Subscriber Export Tool** in [Subscriber Export Tool](#).

Installing of the Subscriber Export Tool on Multiple Sites

Note

Follow this section to fresh install the Subscriber Export Tool on two-sites.

Before installing the Subscriber Export Tool on site 2, you must have the Subscriber Export Tool installed in site 1.

If the Subscriber Export Tool needs to be deployed in site 2, ensure no export tool jobs are running in site 1 until the Subscriber Export Tool is deployed in site 2.

1. Perform the following steps to install the Subscriber Export Tool in multiple sites:
 - a. Make sure the following tables are available across all the sites before installing the Subscriber Export Tool on site 2.
 - EXPORT_BATCH_STEP_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_SEQ
 - EXPORT_BATCH_STEP_EXECUTION_CONTEXT
 - EXPORT_BATCH_JOB_EXECUTION_CONTEXT
 - EXPORT_BATCH_STEP_EXECUTION
 - EXPORT_BATCH_JOB_EXECUTION_PARAMS
 - EXPORT_BATCH_JOB_EXECUTION
 - EXPORT_BATCH_JOB_INSTANCE
 - b. If tables are missing from one of the sites, run the following commands from the site where tables exist. Perform georeplication recovery if there is replication failure:

Note

Do not run the commands if tables are available across all the sites.

- DROP TABLE IF EXISTS EXPORT_BATCH_STEP_EXECUTION_SEQ;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION_SEQ;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_SEQ;
 - DROP TABLE IF EXISTS EXPORT_BATCH_STEP_EXECUTION_CONTEXT;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION_CONTEXT;
 - DROP TABLE IF EXISTS EXPORT_BATCH_STEP_EXECUTION;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION_PARAMS;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_INSTANCE;
- c. Install the Subscriber Export Tool on site 2, refer to [Fresh Installation of the Subscriber Export Tool](#).
 - d. Ensure that the following tables are replicated across all the sites after Subscriber Export Tool is deployed on site 2:
 - EXPORT_BATCH_STEP_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_SEQ
 - EXPORT_BATCH_STEP_EXECUTION_CONTEXT
 - EXPORT_BATCH_JOB_EXECUTION_CONTEXT
 - EXPORT_BATCH_STEP_EXECUTION

- EXPORT_BATCH_JOB_EXECUTION_PARAMS
- EXPORT_BATCH_JOB_EXECUTION
- EXPORT_BATCH_JOB_INSTANCE

Reinstallation of the Subscriber Export Tool on Multiple Sites

① Note

Follow this section if:

1. You are reinstalling the Subscriber Export Tool from an older release version to higher release version and vice versa.
2. You are reinstalling the Subscriber Export Tool in the same release version.

1. Perform the following steps to reinstall the Subscriber Export Tool in multiple sites.
 - a. Run the following command to uninstall the Subscriber Export Tool from all the sites.

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

- b. Install the Subscriber Export Tool on two-sites. To identify the order of Subscriber Export Tool installation, see "Upgrade of Multiple Site Deployments" section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
2. Ensure that the following tables are available across all the sites before installing the Subscriber Export Tool on site 1.
 - EXPORT_BATCH_STEP_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_SEQ
 - EXPORT_BATCH_STEP_EXECUTION_CONTEXT
 - EXPORT_BATCH_JOB_EXECUTION_CONTEXT
 - EXPORT_BATCH_STEP_EXECUTION
 - EXPORT_BATCH_JOB_EXECUTION_PARAMS
 - EXPORT_BATCH_JOB_EXECUTION
 - EXPORT_BATCH_JOB_INSTANCE
 3. If tables are missing from one of the sites, run the following commands from the site where tables exist. Perform georeplication recovery if there is replication failure:

① Note

Do not run the commands if tables are available across all the sites.

- DROP TABLE IF EXISTS EXPORT_BATCH_STEP_EXECUTION_SEQ;
- DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION_SEQ;
- DROP TABLE IF EXISTS EXPORT_BATCH_JOB_SEQ;

- DROP TABLE IF EXISTS EXPORT_BATCH_STEP_EXECUTION_CONTEXT;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION_CONTEXT;
 - DROP TABLE IF EXISTS EXPORT_BATCH_STEP_EXECUTION;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION_PARAMS;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_EXECUTION;
 - DROP TABLE IF EXISTS EXPORT_BATCH_JOB_INSTANCE;
4. Install the Subscriber Export Tool on site 1, refer to [Fresh Installation of the Subscriber Export Tool](#).

Note

Make sure that there are no export tool job running on site1 until the Subscriber Export Tool is deployed on site 2.

5. Ensure that the following tables are replicated across all the sites after the Subscriber Export Tool is installed on site 1:
- EXPORT_BATCH_STEP_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_EXECUTION_SEQ
 - EXPORT_BATCH_JOB_SEQ
 - EXPORT_BATCH_STEP_EXECUTION_CONTEXT
 - EXPORT_BATCH_JOB_EXECUTION_CONTEXT
 - EXPORT_BATCH_STEP_EXECUTION
 - EXPORT_BATCH_JOB_EXECUTION_PARAMS
 - EXPORT_BATCH_JOB_EXECUTION
 - EXPORT_BATCH_JOB_INSTANCE
6. Perform the steps from [2](#) to [5](#) to install the Subscriber Export Tool on site 2.

2.21.6 Customizing Subscriber Export Tool

The following table lists the user parameters for installing and customizing the subscriber export tool.

Table 2-31 Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.dockerRegistry	Docker registry from where the images are pulled	ocudr-registry.us.abc.com:5000/ocudr	NA	NA
global.mysql.dbServiceName	Database service to connect	mysql-connectivity-service.occne-infra	NA	NA
global.mysql.port	Port for database service connection	3306	NA	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.serviceAccountName	The entire pod should be associated with the serviceaccount used for ocudr deployment.	ocudr-serviceaccount	NA	NA
preInstall.image.name	Image name for the nudr-export-tool-prehook pod which will take care of DB and table creation for UDR deployment.	nudr_export_tool_pre_install_hook	NA	NA
preInstall.image.tag	Image version for nudr-export-tool-pre-install-hook pod image	25.1.201	NA	NA
preInstall.config.logLevel	Log level for preinstall hook pod	WARN	Possible Values - WARN INFO DEBUG	NA
postUpgrade.image.name	Image name for the nudr-export-tool-post-upgrade pod which will take care of DB and table creation for UDR deployment.	nudr_export_tool_post_upgrade_hook	NA	NA
postUpgrade.image.tag	Image version for nudr_export_tool_post_upgrade_hook pod image	25.1.201	NA	NA
postUpgrade.config.logLevel	Log level for postUpgrade hook pod	WARN	Possible Values - WARN INFO DEBUG	NA
global.hookJobResources.limits.cpu	CPU limit for pods created k8s hooks/jobs created as part of UDR installation. Applicable for helm test job as well.	1	NA	NA
global.hookJobResources.limits.memory	Memory limit for pods created k8s hooks/jobs created as part of UDR installation. Applicable for helm test job as well.	1Gi	NA	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.hookJobResources.limits.ephemeral-storage	Ephemeral Storage limits for hook pods	1024Mi	Unit MB	NA
global.hookJobResources.requests.cpu	CPU requests for pods created k8s hooks/jobs created as part of UDR installation. Applicable for helm test job as well.	0.5	NA	The cpu to be allocated for hooks during deployment
global.hookJobResources.requests.memory	Memory requests for pods created k8s hooks/jobs created as part of UDR installation. Applicable for helm test job as well.	0.5Gi	NA	The memory to be allocated for hooks during deployment
global.hookJobResources.requests.ephemeral-storage	Ephemeral Storage request for hook pods	72Mi	Unit MB	NA
global.hooksCommonConfig.tolerations	When global.tolerationSetting is ENABLED configure tolerations here.	[]	Example: tolerations: - key: "key1" - operator: "Equal" - value: "value1" - effect: "NoSchedule"	NA
global.hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: - v1 - v2	NA
global.hooksCommonConfig.nodeSelector.nodeKey	NodeSelector key configuration for UDR hooks, This excludes the common service hooks. When hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration will come into use.	' '	Valid key for a label for a particular node	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.hooksCommonConfig.nodeSelector.nodeValue	NodeSelector Value configuration for UDR hooks, This excludes the common service hooks NodeSelector Value configuration for UDR hooks, This excludes the common service hooks	' '	Valid value pair for the above key for a label for a particular node	NA
global.hooksCommonConfig.nodeSelector	NodeSelector configuration when hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion at micro service level is set to v2 Uncomment and use this configuration when required or keep this commented.	{ }	Valid key value pair matching a node for nodeSelection by a pod	
dbCredSecretName	Database Credential Secret Name	ocudr-secrets	NA	NA
configServerEnable	Enable configuration updates via config REST APIs	true	true or false	NA
customExtension.allResources.labels	Custom Annotations that needs to be added to all the OCUDR k8s resources	null	NA	NA
customExtension.allResources.annotations	Custom Annotations that needs to be added to all the OCUDR k8s resources	null	NA	ASM related annotations to be added under ASM Specific Configuration section
customExtension.lbServices.labels	Custom Labels that needs to be added to OCUDR Services that are considered as Load Balancer type	null	NA	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
customExtension.lbServices.annotations	Custom Annotations that needs to be added to OCUDR Services that are considered as Load Balancer type	null	NA	NA
customExtension.lbDeployments.labels	Custom Labels that needs to be added to OCUDR Deployments that are associated to a Service which is of Load Balancer type	null	NA	NA
customExtension.lbDeployments.annotations	Custom Annotations that needs to be added to OCUDR Deployments that are associated to a Service which is of Load Balancer type	null	NA	ASM related annotations to be added under ASM Specific Configuration section
customExtension.nonlbServices.labels	Custom Labels that needs to be added to OCUDR Services that are considered as not Load Balancer type	null	NA	NA
customExtension.nonlbServices.annotations	Custom Annotations that needs to be added to OCUDR Services that are considered as not Load Balancer type	null	NA	NA
customExtension.nonlbDeployments.labels	CustomLabels that needs to be added to OCUDR Deployments that are associated to a Service which is not of Load Balancer type	null	NA	NA
customExtension.nonlbDeployments.annotations	Custom Annotations that needs to be added to OCUDR Deployments 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

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainers	Configuration to enable debug tool container	DISABLED	DISABLED or ENABLED	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainerTpl	The template for extraContainer will be defined in this configuration. For now we have the debugContainer template defined here. Applicable for all microservices under ocudr umbrella.	<pre> - command: - /bin/sleep - infinity image: {{ .Values.global.dockerRegistry }}/ debug-tools:25.1.203 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" </pre>	NA	NA

Table 2-31 (Cont.) Subscriber Export Tool 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		
extraContainersVolumesTpl	Debug container volume template	- name: debug-tools-dir emptyDir: medium: Memory sizeLimit: "4Gi"	NA	NA
image.name	Docker Image name	nudr_export_tool	NA	NA
image.tag	Tag of Image	25.1.201	NA	NA
image.pullPolicy	This setting will tell if image need to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
createExportToolPVC.name	Name of PVC	exporttoolpersisten tclaim	NA	NA
createExportToolPVC.storageClassName	Storage class name for PVC	standard	Possible Values - standard	NA
createExportToolPVC.storage	Storage allocation for PVC	10Gi	NA	NA
createExportToolPVC.pathLocation	Path mounted on PVC for export file generation	/home/udruser/ export	NA	NA
createExportToolPVC.fileName	Fixed file name suffix for export dump files generated	SLFTestSegment	NA	NA
createExportToolPVC.fileExtension	Export file extension	.txt	.csv/.txt	NA
createExportToolPVC.fieldDelimiter	Delimiter used in dump files to separate out different fields	,	NA	NA
createExportToolPVC.multivalueDelimiter	Delimiter used in dump files to separate out multiple values in a single field		NA	NA
createExportToolPVC.fileSize	File size of the generated CSV files. Beyond this limit there will be a new file generated. Do not change this value, this is for Engineering.	1GB	NA	NA
service.type	UDR service type	ClusterIP	Possible Values- ClusterIP NodePort LoadBalancer	The kubernetes service type for exposing UDR deployment. Recommends to be set as ClusterIP default value always.
service.port.http	HTTP port	5001	NA	The http port to be used in notify service to receive signals from nudr-notify-service pod.
service.port.management	bulk import Management port	9000	NA	The actuator management port to be used for notify service.

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
service.customExtension.labels	Custom Labels that need to be added to nudr-diameterproxyspecific Service.	null	NA	This can be used to add custom label(s) to nudr-diameterproxy Service
service.customExtension.annotations	Custom Annotations that need to be added to nudr-diameterproxy specific Services.	null	NA	This can be used to add custom annotation(s) to nudr-diameterproxy Service
deployment.replicaCount	Number of replicas for export tool	1	NA	NA
deployment.customExtension.labels	Custom Labels that need to be added to nudr-diameterproxy specific Deployment.	null	NA	This can be used to add custom label(s) to nudr-diameterproxy Deployment
deployment.customExtension.annotations	Custom Annotations that need to be added to nudr-diameterproxyspecific Deployment	null	NA	This can be used to add custom annotation(s) to nudr-diameterproxy Deployment
hikari.poolsize	Mysql Connection pool size	10	NA	The hikari pool connection size to be created at start up
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
hikari.queryTimeout	Query timeout for a single database query. If you set the value to -1 there will be no timeout for database queries	-1	Unit: Seconds	NA
logging.level.root	Log Level	WARN	Possible Values - WARN INFO DEBUG	Log level of the export tool pod

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
logging.level.ocudr	Application Log Level	WARN	Possible Values - WARN INFO DEBUG	App Log level of the export tool pod
resources.requests.cpu	Cpu Allotment for nudr-notify-service pod	2	NA	The cpu to be allocated for notify service pod during deployment
resources.requests.memory	Memory allotment for nudr-bulk-import pod	2Gi	NA	The memory to be allocated for nudr-bulk-import pod during deployment
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	NA
resources.limits.memory	Memory allotment limitation	2Gi	NA	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
readinessProbe.initialDelaySeconds	Configurable wait time before performing the first readiness probe by the kubelet 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.	30	Not Applicable Unit: Seconds	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
readinessProbe.periodSeconds	Time interval for every readiness probe check. 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.	3	Not Applicable Unit: Seconds	NA
livenessProbe.initialDelaySeconds	Configurable wait time before performing the first liveness probe by the kubelet. 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.	30	Not Applicable Unit: Seconds	NA
livenessProbe.periodSeconds	Time interval for every liveness probe check. 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.	3	Not Applicable Unit: Seconds	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 will let it be dependent on what is configured in the global level

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
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	Do not change this value unless it is different from the value configured in global section reference
istioSidecarReady Url	Readiness url configurable for side car, referred to global section istioSidecarReady Url 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
ocudrReleaseName	Release Name of OCUDR, To form config-server/ocudr endpoint fqdn	ocudr	NA	NA
dbRetry.maxRetry	Number of DB transaction retries for failed DB transactions	3	NA	Do not change the default values. Used for Engineering purpose
dbRetry.interval	Interval for each transaction attempt during retries	4	NA	Do not changes default values. Used for Engineering purpose
tolerations	When tolerationsSetting is ENABLED configure tolerations here	[]	NA	Example: tolerations: - key: "key1" operator: "Equal" value: "value1" effect: "NoSchedule"
helmBasedConfigurationNodeSelector ApiVersion	Node Selector API version setting	v1	Allowed Values: v1 v2	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeSelector.node Key	Micro service level NodeSelector key configuration. When helmBasedConfigurationNodeSelector ApiVersion is set to v1 then this configuration will come into use. The 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.node Value	Global level NodeSelector Value configuration. When helmBasedConfigurationNodeSelector ApiVersion is set to v1 then this configuration will come into use. The 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 helmBasedConfigurationNodeSelector ApiVersion is set to v2. Uncomment and use this configuration when required or keep this commented.	{}		Valid key value pair matching a node for nodeSelection by a pod
periodicDumpFrequency	Frequency for periodic dump generation. By default only one time is set	ONE_TIME	ONE_TIME/DAILY/WEEKLY/MONTHLY	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
startTime	Start time for dump generation. If left empty, then now will be considered for dump generation. Date is required, so for WEEKLY dump the date is used for identifying day in the week and for subsequent weeks the same day is used	14/09/2022 17:32:10	Valid time in the format as shown in default value/ Empty ("")	NA
maxDumps	Max dumps that will be retained in the PVC storage. Rest of the dumps older in age will be cleared.	5	[1 - 5]	NA
numberOfSubscribers	Internal Engineering Configuration. Do not change	1000	NA	NA
batch.corePoolSize	Internal Engineering Configuration. Do not change	5	NA	NA
batch.maxPoolSize	Internal Engineering Configuration. Do not change	5	NA	NA
batch.queueCapacity	Internal Engineering Configuration. Do not change	30000	NA	NA
sftpDetails.host	Remote Host IP	10.75.229.18	NA	NA
sftpDetails.port	Remote Host Port	22	NA	NA
sftpDetails.username	Remote Host Username	cloud-user	NA	NA
sftpDetails.path	Path of Remote host server where the exported files are stored	/home/cloud-user/ocudr/	NA	Do not delete the backslash / in the default value. It is mandatory to include the backslash. This is also applicable during the REST API operation.

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
sftpDetails.sftpExportEnabled	Indicates if the sftp export is enabled or not	true	NA	NA
sftpDetails.deleteFromPvc	Indicates whether files should be deleted from PVC after export	true	NA	NA
sftpDetails.pollingInterval	Interval at which files are polled for sftp export in minutes	1	NA	NA
sftpDetails.secrets.privatekey.name	Private key name of the remote server in secrets	ocudr-ssh-private-key	NA	NA
sftpDetails.secrets.publickey.name	Public key name of remote server in secrets	ocudr-ssh-public-key	NA	NA
exportMode	Indicates SLF or EIR subscribers export mode. Values of export must be CSV_EXPORT(SLF Mode)/PDBI_EXPORT(EIR Mode)/EXML_EXPORT(4G Policy subscriber export)/POLICY_DATA_EXPORT(Policy Data Subscribers CSV Export).	PDBI_EXPORT	PDBI_EXPORT/ CSV_EXPORT/ EXML_EXPORT/ POLICY_DATA_EXPORT	NA
imsiCount	Applicable for PDBI_EXPORT. The max number of IMSIs as part of a single subscriber that will be exported	5	[1-10]	NA
msisdnCount	Applicable for PDBI_EXPORT. The max number of MSISDNs as part of a single subscriber that will be exported	5	[1-10]	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
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	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	NA	NA
maxConcurrentPushedStreams	Maximum number of concurrent requests that is pushed per destination	6000	NA	NA
maxRequestsQueuedPerDestination	Maximum number of requests that is 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	NA	NA
idleTimeout	Client connection idle timeout	0	NA	NA
dnsRefreshDelay	Server DNS refresh delay	120000	NA	NA
pingDelay	Server ping delay	60	NA	NA
connectionFailureThreshold	Client connection failure threshold	10	NA	NA

Table 2-31 (Cont.) Subscriber Export Tool 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/false	NA
subscriberRangeEnabled	Flag to enable or disable the range based subscriber export. Default is set to false for full range of 4G policy subscriber data export.	false	true/false	NA
keyType	To export the range based subscriber data with keys. The two key type are IMSI and MSISDN.	msisdn	msisdn/imsi	NA
subscriberRanges	Configures the start and end ranges that is given to export the data. This works only if subscriberRangeEnabled parameter is enabled.	1111111130-1111111140, 1111111150-1111111160	NA	NA
newFilePerRangeEnabled	When enabled creates a separate EXML file for each range. Default is set to false.	false	true/false	NA
sftpDetails.fileSizeCheck	Checks the file size of the remote server once the file is successfully transferred to remote location.	false	true/false	NA
sftpDetails.transferProtocol	Transfers the files to remote location. the default protocol uses SFTP	sftp	sftp or scp	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	30s	Unit: Seconds	NA
fileStatusInterval	It is the file status update interval. This is an engineering configurations.	30	Unit: Seconds	NA

Table 2-31 (Cont.) Subscriber Export Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
jobStatusInterval	It is the Jib status update interval. This is an engineering configurations.	30	Unit: Seconds	NA
policyDataResources	It is the type of data that needs to be exported for the policy data during POLICY_DATA_EXPORT mode. All types of data, such as sm-data, am-data, and ue-policy-set can be exported. You can configure the data in CSV format in the POLICY_DATA_EXPORT mode to export the policy data.	sm-data, am-data, ue-policy-set	Comma separated resource names. allowed values are sm-data, am-data, ue-policy-set. One of the following values are mandatory: • am-data • am-data, ue-policyset • sm-data, am-data	NA

To Configure on-demand and schedule periodic dumps on the export tool post the installation, see the **Configuration APIs for UDR Microservices** and **Configuration APIs for Loglevel** sections in the **Configuration API Specification** chapter of the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

The subscriber file export will be scheduled when a REST API is requested by providing the below the parameters. The export will start on the configured **startTime**. Details of the configuration present in the payload are given below:

1. **frequencyOfPeriodicDump**: When the frequencyOfPeriodicDump parameter is configured as ONE_TIME, it will act as ondemand dump and the startTime must be set to a value from the current time
2. **startTime**: Start time of the dump must be configured in day-month-year-hours-minutes-seconds (dd-MM-YYYY hh:mm:ss) and 24 hours format. In case of WEEKLY start time the date is used to identify the day of the week the dump is scheduled
3. **maxNumberOfDumps**: Maximum number of dumps that are retained on the Persistent Volume Claim (PVC). When the maximum limit is reached, the oldest dump is cleared
4. **fixedDumpFileNamePrefix**: Configurable name suffix for each dump file that is generated. This is used along with the timestamp
5. **multiValuedelimiter**: This is used as the delimiter when we have multiple keys of the same type
6. **fieldDelimiter**: Field delimiter is used as the delimiter for each field on the dump file
7. **fileExtension**: Configurable file extension is used for the generated dump files

8. **exportMode**: This is the mode of export. The values of the export must be SLF_EXPORT for SLF Mode, EXML_EXPORT for 4G Subscriber export, POLICY_DATA_EXPORT for policy data subscribers export, and EIR_EXPORT for EIR mode. This field is not editable.
9. **imsiCount**: Configurable imsi count. It is used to write that many imsi column in export file
10. **msisdnCount**: Configurable msisdn count. It is used to write that many msisdn column in export file
11. **remoteUserNam**: Configurable user name of remote host
12. **remotePath**: Configurable path in remote server where exported files is stored
13. **serverHost**: Configurable IP address of remote host
14. **serverPort**: Configurable Port of remote host
15. **sftpExportEnabled**: Used to indicate if the SFTP is enabled or disabled
16. **deleteFromPVC**: Configurable flag to control deletion of files from sftp
17. **pollingInterval**: Polling interval for SFTP functionality measured in minutes
18. **subscriberRangeEnabled**: Flag to enable or disable the range based subscriber export. Default is set to false for full range of 4G policy subscriber data export.
19. **keyType** : This is used to export the range based subscriber data with keys. The two key type are IMSI and MSISDN.
20. **subscriberRanges**: Configures the start and end ranges that is given to export the data.
21. **newFilePerRangeEnabled** : Creates a separate EXML file for each range. Default is set to false.
22. **sftpDetails.fileSizeCheck**: Checks the file size of the remote server once the file is successfully transferred to remote location.
23. **sftpDetails.transferProtocol**: Transfers the files to remote location. The default protocol is SFTP.

2.21.7 Upgrade and Rollback support for Subscriber Export Tool

Upgrade Procedure

Note

Helm Upgrade and Helm rollback of Subscriber Export Tool is not supported.

Helm Upgrade of Subscriber Export Tool is not supported. You must uninstall the current version Subscriber Export Tool and install the higher version. For more information, see [Installing Subscriber Export Tool](#).

Rollback Procedure

Helm rollback of Subscriber Export Tool is not supported. Perform the following steps to rollback Subscriber Export Tool:

Note

UDR and the Subscriber Export Tool must be rolled back together.

- Uninstall the current version of the Subscriber Export Tool.
- Install the older version of the Subscriber Export Tool. For more information, see [Installing Subscriber Export Tool](#).
- Perform the rollback procedure for UDR. For more information, see Rolling Back UDR chapter in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

2.21.8 Monitoring the Subscriber Export Tool Status

This section provides details of monitoring subscriber export tool status.

Monitoring the Subscriber Export Tool Operation

You can monitoring the export operation is supported using REST API and CNC Console. It also supports monitoring the status of exported data and overall export operation through CNC Console.

The export file status has the following information:

- `fileName`: Export file name
- `startTime`: The start time of subscriber export to the particular file
- `completeTime`: Current File dumping completion time
- `filePath`: The path where the file is stored on the container, and the same is mounted on the PVC
- `status`: Status are INPROGRESS or COMPLETED
- `jobId`: Unique ID for Export

The overall export operation status has the following information:

- `subscriberCount`: 220
- `exportedSubscriberCount`: 220
- `jobProgress`: 100 (In percentage)
- `jobId`: Unique ID for Export
- `status`: RUNNING/COMPLETED/SKIPPED/FAILED

Note

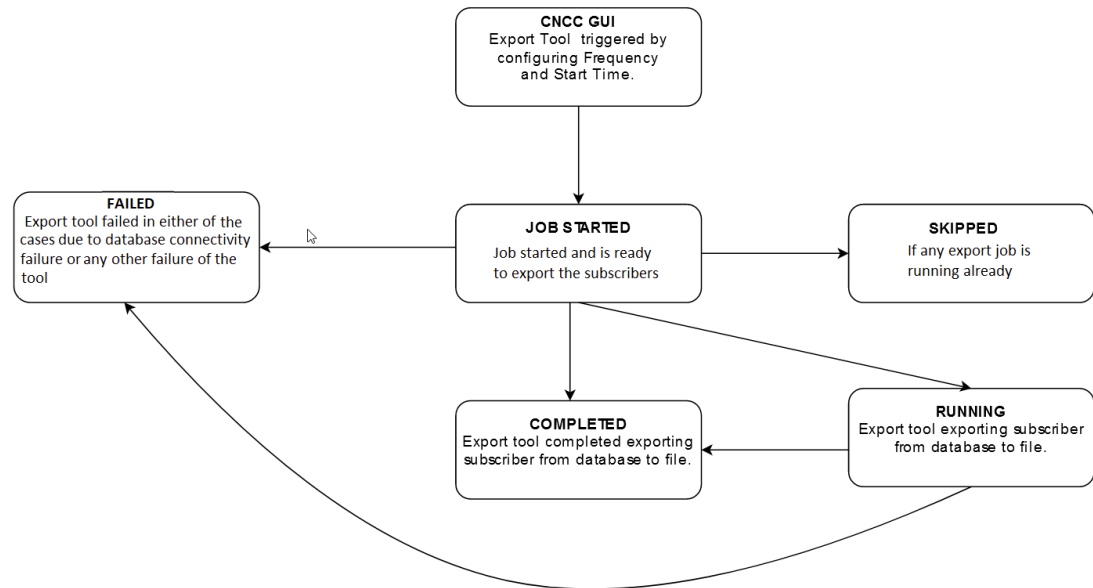
The subscriber export tool's cleaning mechanism is enhanced, and the database and PVC will sync when any status data or file is deleted. If you enable the `deleteFromPVC` parameter, it will delete the exported file after it is transferred to the remote server and remove the status data from the database.

Subscriber Export Tool Job Status

When the subscriber export tool is triggered by configuring the frequency and start time in the CNC Console, the subscriber export tool initiates exporting the subscribers based on the schedule configured. If another export is running at that time, the export will be skipped. If the subscriber data is not found by the subscriber export tool, the export status will be displayed as completed. If the subscriber data is available, the export will start and display the complete state when the export is successful. The export will be in failed state, if the below scenario occurs:

- If any resource is not found, for example: database connectivity.
- If the subscriber export tool fails to connect to the database.
- In case of failure of the subscriber export tool.

Figure 2-18 Subscriber Export Tool Export Status

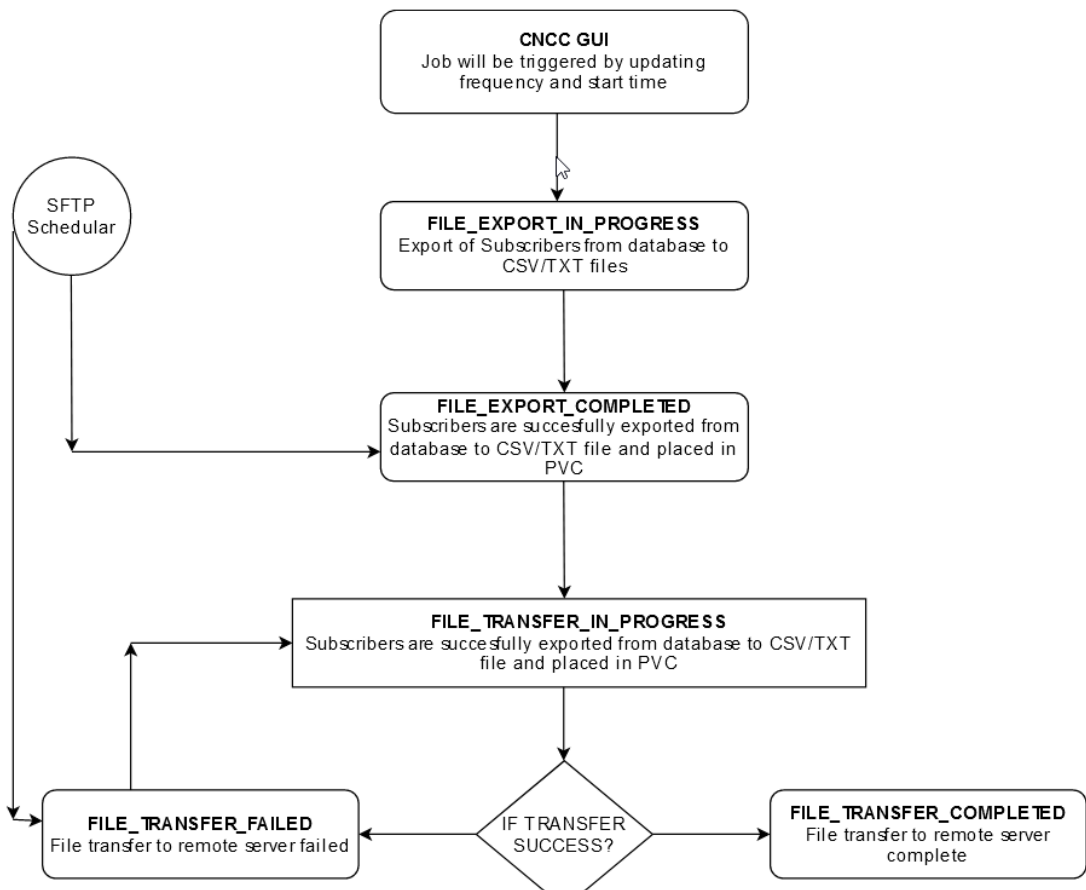


Subscriber Export Tool File Status

The following are the file status during export of subscriber data file:

- When the subscriber export tool is triggered by configuring the frequency and start time using the CNC Console, the file status shows `FILE_EXPORT_IN_PROGRESS`.
- When the subscriber data is exported from database to CSV or TXT file, the file status shows `FILE_EXPORT_COMPLETE`.
- When the file export is completed, the SFTP scheduler exports the subscriber data from the database to CSV or TXT file and the file is placed in the PVC. The file status shows `FILE_TRANSFER_IN_PROGRESS`.
- If the transfer is successful, file status shows `FILE_TRANSFER_COMPLETED`.
- If the file transfer is failed, file status shows `FILE_TRANSFER FAILED` and the SFTP scheduler picks up the file again for transfer and the file status shows `FILE_TRANSFER_IN_PROGRESS`.

Figure 2-19 Subscriber Export Tool File Status



2.22 Subscriber Activity Logging

UDR supports multiple log levels which the operator can set based on the requirement. The default log level enabled is ERROR or WARN. Subscriber Activity Logging enhances the logging functionality of UDR to support subscriber specific logging. This feature enables the operators to log the subscriber activity for the configured list of subscribers. A list of subscribers for which the logging is required can be defined and this is controlled irrespective of the log level that is enabled for UDR. You can collect the message exchanges, subscriber specific decision making, and logic handling for specific subscribers.

The subscriber activity logs are considered for the following cases:

- 5G signaling traffic - SBI
- Diameter traffic - Sh Interface
- Provisioning traffic

Note

You can enable Subscriber Activity Logging with up to 100 keys.

To Enable

You can enable or disable subscriber activity logging using CNC Console or Helm Configuration.

- You can disable or enable the feature by setting the `subscriberActivityEnabled` parameter to true or false in the global section of the `ocudr-custom-values.yaml` file. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- You can disable or enable the feature by setting the `subscriberActivityEnabled` parameter to true or false in the [Global Configurations](#) in the CNC Console

Flag to disable or enable the feature is given in the global section of the `ocudr-custom-values.yaml` file.

Note

When the requested subscriber logging is enabled the corresponding subscriber Identifier must be cleared because additional logging can impact performance.

Global section

```
subscriberActivityEnabled:false
```

Configure using CNC Console:

You can configure different subscriber identifiers for each keys. Subscriber activity is enabled only for values which are configured. For more information, see [Global Configurations](#) in the CNC Console.

```
"subscriberIdentifiers": {
    "msisdn": ["1111111123", "1111111124", "1111111125", "1111111111"],
    "imsi": ["222222221", "222222222", "222222223"],
    "nai": ["333333331", "333333332", "333333333"],
    "extid": ["233333331"]
}
```

Configure using REST API:

You can perform the PUT operation on the `http://10.75.214.219:32113/nudr-config/v1/udr.global.cfg/GLOBAL` to configure the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "ingressHttpPort": "80",
  "nfInstanceId": "5a7bd676-ceed-44bb-95e0-f6a55a328b03",
  "snssai": "2-FFFFF",
  "ingressHttpsPort": "",
  "dnn": "dnn1",
  "dbServiceName": "mysql-connectivity-service.occne-asmndb",
  "autoCreate": true,
  "udsfEnabled": false,
```

```

    "vsaDefaultBillingDay": "1",
    "sbiCorrelationInfoEnable": false,
    "subscriberActivityEnabled": true,
    "consumerNF": "PCF,UDM,NEF",
    "subscriberIdentifiers": {
        "nai": [],
        "imsi": ["0908338283"],
        "extid": [],
        "msisdn": ["1234567890", "1234567891"]
    },
    "udrServices": "All"
}

```

Sample logs

Below is the log snippet when subscriber automatically create is disabled for signaling:

```

{"instant":
{"epochSecond":1658758894,"nanoOfSecond":319666506},"thread":"XNIO-1
task-1","level":"WARN","loggerName":"ocudr.udr.services.common.metrics.Subscri
berActivityLogger","message":"Subscriber Activity
Logging :msisdn-1111111124 : Incoming Request received
","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.AbstractLogger
","threadId":45,"threadPriority":5,"messageTimestamp":"2022-07-25T14:21:34.319
+0000","application":"redy1","engVersion":"22.2.3012","mktgVersion":"22.2.0.0
.0","microservice":"redy1-nudr-
drservice","vendor":"Oracle","subscriberId":"msisdn-1111111124","resourceId":"
nudr-dr/v2/policy-data/ues/ue-policy-set","ocLogId":"1658758874753_269_reddy1-
ingressgateway-sig-666c94f6b8-lnxhw","requestType":"PUT"}
{"instant":
{"epochSecond":1658758894,"nanoOfSecond":398003490},"thread":"XNIO-1
task-1","level":"WARN","loggerName":"ocudr.udr.services.common.metrics.Subscri
berActivityLogger","message":"Subscriber Activity
Logging :msisdn-1111111124 : schema validation
success","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.Abstrac
tLogger","threadId":45,"threadPriority":5,"messageTimestamp":"2022-07-25T14:21
:34.398+0000","application":"redy1","engVersion":"22.2.3012","mktgVersion":"2
2.2.0.0.0","microservice":"redy1-nudr-
drservice","vendor":"Oracle","subscriberId":"msisdn-1111111124","resourceId":"
nudr-dr/v2/policy-data/ues/ue-policy-set","ocLogId":"1658758874753_269_reddy1-
ingressgateway-sig-666c94f6b8-lnxhw","requestType":"PUT"}
{"instant":
{"epochSecond":1658758894,"nanoOfSecond":610396593},"thread":"XNIO-1
task-1","level":"WARN","loggerName":"ocudr.udr.services.common.metrics.Subscri
berActivityLogger","message":"Subscriber Activity
Logging :msisdn-1111111124 : Subscriber not exist in 5G
","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.AbstractLogger
","threadId":45,"threadPriority":5,"messageTimestamp":"2022-07-25T14:21:34.610
+0000","application":"redy1","engVersion":"22.2.3012","mktgVersion":"22.2.0.0
.0","microservice":"redy1-nudr-
drservice","vendor":"Oracle","subscriberId":"msisdn-1111111124","resourceId":"
nudr-dr/v2/policy-data/ues/ue-policy-set","ocLogId":"1658758874753_269_reddy1-
ingressgateway-sig-666c94f6b8-lnxhw","requestType":"PUT"}
{"instant":
{"epochSecond":1658758894,"nanoOfSecond":626709333},"thread":"XNIO-1
task-1","level":"WARN","loggerName":"ocudr.udr.services.common.metrics.Subscri

```

```
berActivityLogger", "message": "Subscriber Acitivity
Logging :msisdn-1111111124 : Auto create is disabled- Subscriber not
found", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.AbstractL
ogger", "threadId": 45, "threadPriority": 5, "messageTimestamp": "2022-07-25T14:21:3
4.626+0000", "application": "redy1", "engVersion": "22.2.3012", "mktgVersion": "22.
2.0.0.0", "microservice": "redy1-nudr-
drservice", "vendor": "Oracle", "subscriberId": "msisdn-1111111124", "resourceId": "
nudr-dr/v2/policy-data/ues/ue-policy-set", "ocLogId": "1658758874753_269_reddy1-
ingressgateway-sig-666c94f6b8-lnxhw", "requestType": "PUT"}
```

Below is the log snippet when sequence number for state entity is out of synchronize for diameter:

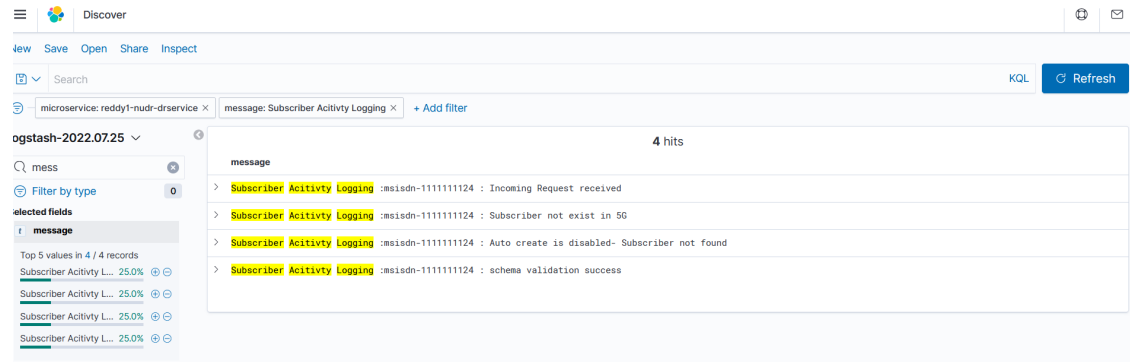
```
{"instant":
{"epochSecond":1658827381,"nanoOfSecond":998796077},"thread":"DiamMsgBuffer:Wo
rkerThread-5","level":"WARN","loggerName":"ocudr.uds.services.common.metrics.S
ubscriberActivityLogger","message":"Subscriber Acitivity
Logging :imsi-111111111111 : Received PUR
msg", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLog
ger", "threadId": 5, "threadPriority": 5, "messageTimestamp": "2022-07-26T09:23:01.9
98+0000", "application": "ocudr", "engVersion": "22.2.40", "mktgVersion": "22.2.0.0.
0", "microservice": "ocudr-nudr-
diameterproxy", "vendor": "Oracle", "subscriberId": "imsi-111111111111", "ocLogId":
"1658827381997_5_ocudr-nudr-diameterproxy-55f6b54cb6-
cdwx5", "requestType": "PUR", "originHost": "Route-Record (282,M,l=27) =
diamcli1.oracle.com"}
{"instant":
{"epochSecond":1658827381,"nanoOfSecond":999854337},"thread":"DiamMsgBuffer:Wo
rkerThread-5","level":"WARN","loggerName":"ocudr.uds.services.common.metrics.S
ubscriberActivityLogger","message":"Subscriber Acitivity
Logging :imsi-111111111111 : Data-Reference validation
successful", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.Abstr
actLogger", "threadId": 5, "threadPriority": 5, "messageTimestamp": "2022-07-26T09:
23:01.999+0000", "application": "ocudr", "engVersion": "22.2.40", "mktgVersion": "22
.2.0.0.0", "microservice": "ocudr-nudr-
diameterproxy", "vendor": "Oracle", "subscriberId": "imsi-111111111111", "ocLogId":
"1658827381997_5_ocudr-nudr-diameterproxy-55f6b54cb6-
cdwx5", "requestType": "PUR", "originHost": "Route-Record (282,M,l=27) =
diamcli1.oracle.com"}
{"instant":
{"epochSecond":1658827382,"nanoOfSecond":3210128},"thread":"DiamMsgBuffer:Wo
rkerThread-5","level":"WARN","loggerName":"ocudr.uds.services.common.metrics.Sub
scriberActivityLogger","message":"Subscriber Acitivity
Logging :imsi-111111111111 : Schema Validation
Successful", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.Abstr
actLogger", "threadId": 5, "threadPriority": 5, "messageTimestamp": "2022-07-26T09:
23:02.003+0000", "application": "ocudr", "engVersion": "22.2.40", "mktgVersion": "22
.2.0.0.0", "microservice": "ocudr-nudr-
diameterproxy", "vendor": "Oracle", "subscriberId": "imsi-111111111111", "ocLogId":
"1658827381997_5_ocudr-nudr-diameterproxy-55f6b54cb6-
cdwx5", "requestType": "PUR", "originHost": "Route-Record (282,M,l=27) =
diamcli1.oracle.com"}
{"instant":
{"epochSecond":1658827382,"nanoOfSecond":13014813},"thread":"DiamMsgBuffer:Wo
rkerThread-5","level":"WARN","loggerName":"ocudr.uds.services.common.metrics.Su
```

```

bscriberActivityLogger", "message": "Subscriber Acitivity
Logging :imsi-11111111111 : inserting or updating data into 5G
UDR", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLog
ger", "threadId": 5, "threadPriority": 5, "messageTimestamp": "2022-07-26T09:23:02.0
13+0000", "application": "ocudr", "engVersion": "22.2.40", "mktgVersion": "22.2.0.0
", "microservice": "ocudr-nudr-
diameterproxy", "vendor": "Oracle", "subscriberId": "imsi-11111111111", "ocLogId":
"1658827381997_5_ocudr-nudr-diameterproxy-55f6b54cb6-
cdwx5", "requestType": "PUR", "originHost": "Route-Record (282,M,l=27) =
diamcli1.oracle.com"}
{"instant":
{"epochSecond": 1658827382, "nanoOfSecond": 16845880}, "thread": "DiamMsgBuffer:Wor
kerThread-5", "level": "WARN", "loggerName": "ocudr.udr.services.common.metrics.Su
bscriberActivityLogger", "message": "Subscriber Acitivity
Logging :imsi-11111111111 : caught exception from dao
operation", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.Abstr
actLogger", "threadId": 5, "threadPriority": 5, "messageTimestamp": "2022-07-26T09:2
3:02.016+0000", "application": "ocudr", "engVersion": "22.2.40", "mktgVersion": "22.
2.0.0.0", "microservice": "ocudr-nudr-
diameterproxy", "vendor": "Oracle", "subscriberId": "imsi-11111111111", "resultCod
e": "409", "ocLogId": "1658827381997_5_ocudr-nudr-diameterproxy-55f6b54cb6-
cdwx5", "requestType": "PUR", "originHost": "Route-Record (282,M,l=27) =
diamcli1.oracle.com"}
{"instant":
{"epochSecond": 1658827382, "nanoOfSecond": 17614851}, "thread": "DiamMsgBuffer:Wor
kerThread-5", "level": "WARN", "loggerName": "ocudr.udr.services.common.metrics.Su
bscriberActivityLogger", "message": "Subscriber Acitivity
Logging :imsi-11111111111 :
DIAMETER_ERROR_TRANSPARENT_DATA_OUT_OF_SYNC", "endOfBatch": false, "loggerFqcn": "
org.apache.logging.log4j.spi.AbstractLogger", "threadId": 5, "threadPriority": 5, "
messageTimestamp": "2022-07-26T09:23:02.017+0000", "application": "ocudr", "engVer
sion": "22.2.40", "mktgVersion": "22.2.0.0.0", "microservice": "ocudr-nudr-
diameterproxy", "vendor": "Oracle", "subscriberId": "imsi-11111111111", "resultCod
e": "409", "ocLogId": "1658827381997_5_ocudr-nudr-diameterproxy-55f6b54cb6-
cdwx5", "requestType": "PUR", "originHost": "Route-Record (282,M,l=27) =
diamcli1.oracle.com"}
{"instant":
{"epochSecond": 1658827382, "nanoOfSecond": 18276262}, "thread": "DiamMsgBuffer:Wor
kerThread-5", "level": "WARN", "loggerName": "ocudr.udr.services.common.metrics.Su
bscriberActivityLogger", "message": "Subscriber Acitivity
Logging :imsi-11111111111 : request resulCode:
5105", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLo
gger", "threadId": 5, "threadPriority": 5, "messageTimestamp": "2022-07-26T09:23:02.
018+0000", "application": "ocudr", "engVersion": "22.2.40", "mktgVersion": "22.2.0.0
.0", "microservice": "ocudr-nudr-
diameterproxy", "vendor": "Oracle", "subscriberId": "imsi-11111111111", "resultCod
e": "409", "ocLogId": "1658827381997_5_ocudr-nudr-diameterproxy-55f6b54cb6-
cdwx5", "requestType": "PUR", "originHost": "Route-Record (282,M,l=27) =
diamcli1.oracle.com"}

```

Sample log from kibana dashboard is shown below.

Figure 2-20 Sample Kibana Logs**To Observe**

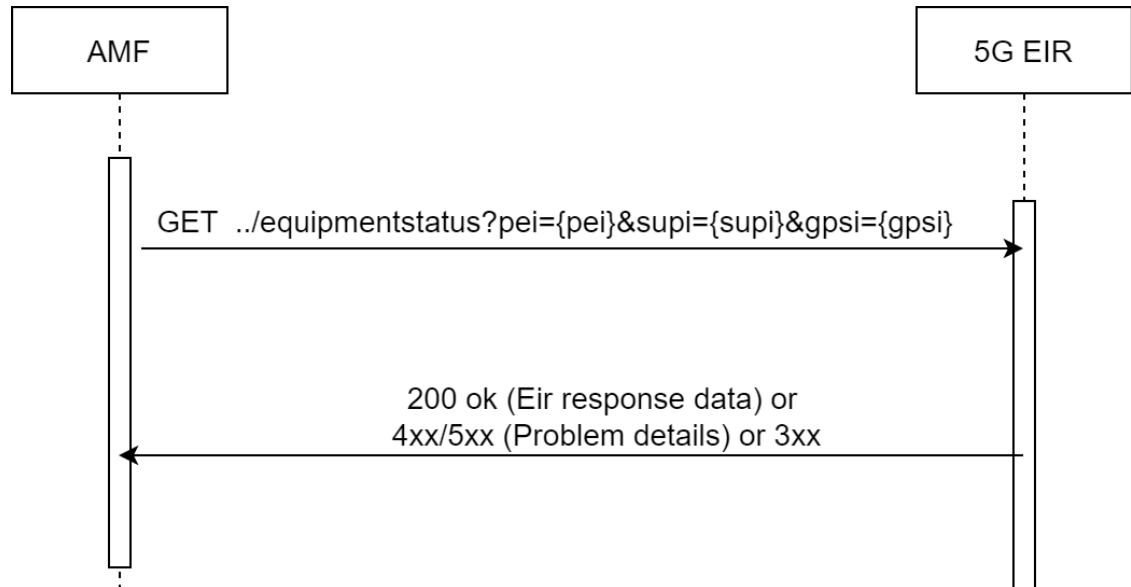
See [UDR Metrics](#) section for more information on metrics.

2.23 Support for 5G EIR Signaling

The Equipment Identity Register (EIR) is a network entity used in Global System for Mobile communication (GSM) networks that stores lists of International Mobile Equipment Identity (IMEI) numbers, which correspond to physical handsets (not subscribers). This feature introduces 5G EIR functionality on the UDR. UDR supports a separate deployment mode for 5G EIR. 5G EIR supports:

- REST/JSON based provisioning of 5G EIR profile
- Provisioning of 5G EIR entries using CNC Console or REST API
- N5g-eir_EquipmentIdentityCheck Service API. N5g-eir is the service based interface as defined by 3GPP.
- NRF registration of the EIR service
- EIR Access Log for blacklisted and greylisted Permanent Equipment Identifier (PEI). 5G EIR supports 14 digit International Mobile Equipment Identity (IMEI).
- Bulk import of 5G EIR subscriber data by using Provisioning Database Interface (PDBI) interface and Comma Separated Value (CSV)

5G EIR provides N5g-eir_Equipment Identity Check service to check the PEI status and determine if the subscriber is allowed to use the equipment. This following diagram depicts the 5G-EIR call flow:

Figure 2-21 Reference Model – 5G-EIR

The NF Service Consumer (for example: AMF) checks the status of the PEI by using the HTTP GET method. The NF Service Consumer sends a GET to the resource representing the PEI equipment status. The request includes the PEI as the query parameter and if required, Subscription Permanent Identifier (SUPI) or Generic Public Subscription Identifier (GPSI) can be included.

When the request is successful, "200 OK" is received with the message body containing the equipment status of the PEI.

If the equipment status of the PEI is not known, 404 Not Found error is returned with the message body containing problem details. The details attribute is set to `ERROR_EQUIPMENT_UNKNOWN`.

EIR Signaling Interface

EIR signaling interface supports HTTP2 and utilizes the following feature:

- The TLS and oauth2 authentication features provided by the ingressgateway-sig microservice.
- The overload control feature provided by the ingressgateway-sig microservice.
- The User-Agent, LCI and OCI, XFCC, and server header support on the ingressgateway-sig microservice.
- The Rate Limiting feature provided by ingressgateway-sig microservice.

EIR Equipment Identity Check

The URL supported for `N5g-eir_EquipmentIdentityCheck` is defined in 3GPP 29.511 specification. For example, `{api-root}/n5g-eir-eic/v1/equipment-status?pei={pei}&supi={supi}&gpsi={gpsi}`

If the PEI and keys received in the EIC request have an exact match in the database, then the corresponding status is filled in the EirResponseData. For example:

```
{
  "status": "WHITELISTED"
}
```

If PEI matches and other keys received in the EIC request is not associated with this PEI, then the response is based on the configuration `nudr-dr.service.defaultResponse`. The configuration is set to `EQUIPMENT_UNKNOWN` by default. The response will be 404 with cause as `ERROR_EQUIPMENT_UNKNOWN` as follows.

```
{
  "title": "Not Found",
  "status": 404,
  "detail": "Provided SUPI/GPSI does not match with PEI association",
  "cause": "ERROR_EQUIPMENT_UNKNOWN",
  "invalidParams": []
}
```

If the PEI is not present, then 404 will be returned with problem details. The sample problem details that will be present in EIC response is given as follows:

```
{
  "title": "Not Found",
  "status": 404,
  "detail": "Mobile Equipment Not Known in the EIR",
  "cause": "ERROR_EQUIPMENT_UNKNOWN",
  "invalidParams": []
}
```

To Install 5G EIR

You can configure UDR, to run in 5G EIR mode using the Helm based configuration. Modify the `ocudr-custom-values.yaml` file with the following configuration parameters to install in EIR mode:

- set value of the `global.udrServices` parameter to `n5g-eir-eic`.
- Set the `appinfo.coreservices` parameters as below.

```
# Service to be monitored by appinfo, we need to add to list as and when
required
#If UDR service is ALL OR nudr-dr use *allordr, use *slf if
udrServices=nudr-group-id-map
core_services:
  #udr: *allordr
  #if EIR mode then comment udr and uncomment below
  5g_eir: *eir
```

- Uncomment EIR appProfiles in `nrf-client` section.

```
#uncomment below appProfiles if EIR mode is enabled
appProfiles= [{"nfInstanceId": "5a7bd676-ceed-44bb-95e0-f6a55a328b03",
```



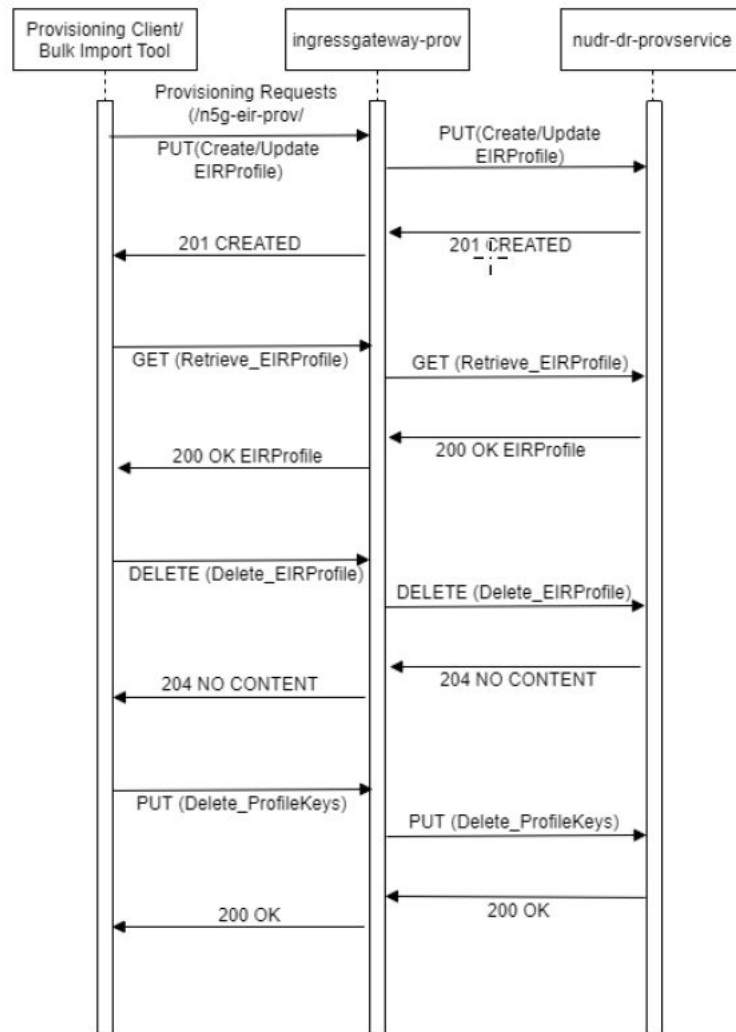
```
"nfStatus": "REGISTERED", "fqdn": "ocudr-
ingressgateway.myudr.svc.cluster.local", "nfType": "5G_EIR",
"allowedNfTypes": ["AMF"], "plmnList": [{ "mnc": "14", "mcc": "310" }],
"priority": 10, "capacity": 500, "load": 0, "locality": "bangalore",
"nfServices": [{ "load": 0, "scheme": "http", "versions":
[{ "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"]}]}
```

For more information about above parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

To support 5G EIR bulk provisioning a new container (pdbiToCsv container) has been added in the bulk import tool. To configure bulk import with this enhancement, see [Subscriber Bulk Import Tool](#).

5G EIR Provisioning

5G EIR provisioning interface is supported on the ingressgateway-prov microservice and utilizes the TLS and oauth2 authentication features provided by the ingressgateway-prov microservice. EIR provisioning interface supports HTTP1.1 and HTTP2. Internally, provisioning requests are routed to nudr-dr-provservice, which creates subscriber records on the EIR database. For more information, see [Subscriber Bulk Import Tool](#).



EIR Access Logs

The attempts made by IMEIs on the black and grey lists to access the network is logged in EIR. The below details are logged:

- Date and Time
- Source
- IMSI
- IMEI
- Status

Note

- The status of the IMEI is indicated as below:
 - IMEI in black list is 1
 - IMEI in grey list is 2
- Access logs are applicable when defaultResponse is used and it is set as GREYLISTED or BLACKLISTED.

The access logs can be filtered on the kibana for a specific time period using EIR access log filter on nudr-drservice logs. EIR access log is flag driven feature. A new configuration, nudr-drservice.accessLogEnabled, is introduced to enable or disable the feature.

When the EIR access log feature is enabled and there is an equipment identity check, the access log is written for subscribers in greylist and blacklist. Sample for EIR access logs is given below.

```
{
  "instant":
  {
    "epochSecond":1656481285, "nanoOfSecond":195874715, "thread": "XNIO-1
    task-1", "level": "ERROR", "loggerName": "ocudr.udr.services.util.EIRAccessLogger"
    , "message": "EIR Access Log: pei-1111111113, supi-1111111112, gpsi-1111111121,
    status:
    2", "endOfBatch": false, "loggerFqcn": "org.apache.logging.log4j.spi.AbstractLogge
    r", "threadId": 44, "threadPriority": 5, "messageTimestamp": "2022-06-29T05:41:25.19
    5+0000", "application": "ocudr", "engVersion": "22.2.30-
    m1", "mktgVersion": "22.2.0.0.0", "microservice": "ocudr-nudr-
    drservice", "vendor": "Oracle", "resourceId": "n5g-eir-eic/v1/equipment-
    status", "ocLogId": "1656481291296_128_ocudr-ingressgateway-sig-65f875b999-
    cc5zd", "requestType": "GET"}
}
```

To Observe

See [UDR Metrics](#) section for more information on metrics.

2.23.1 User Equipment Optimization Support for EIR

As per 3GPP standards, all User Equipment (UE) in the operators network is declared within the EIR database as Whitelisted, Greylisted, or Blacklisted. The UEs that are not declared in the EIR database are categorized as Blacklisted by default. As all UE Permanent Equipment Identifier (PEI) and International Mobile Equipment Identity (IMEI) cannot be provisioned in EIR database. This feature allows the operators to configure a default response for unknown UE as follows:

- 200 OK - WHITELISTED/BLACKLISTED/GREYLISTED
- 404 Not Found - Equipment Not Found

When the PEI or IMEI are received in the GET request but PEI or IMEI are not found in the EIR database, EIR responds with a configured default response. The default response is set as Equipment Not Found. The sample response from the Equipment Identity Check (EIC) is as follows:

```
{
  "title": "Not Found",
}
```

```

    "status": 404,
    "detail": "Equipment not found",
    "cause": "ERROR_EQUIPMENT_UNKNOWN",
    "invalidParams": []
  }

```

Configure using REST API:

You can perform the PUT operation on the `http://10.75.215.124:32523/nudr-config/v1/udr.nudrservice.cfg/DR-SERVICE` to configure the default response. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```

{
  "hikariPoolSize": 25,
  "tracingEnabled": false,
  "subscriptionDataSubscriptionsOnly": false,
  "defaultGroupIdEnabled": false,
  "defaultSlfGroupName": "DefaultGrp",
  "onDemandMigrationEnabled": true,
  "defaultResponse": "EQUIPMENT_UNKNOWN",
  "accessLogEnabled": false,
  "defaultResponseIMEINotFound": "WHITELISTED"
}

```

When the `defaultResponseIMEINotFound` parameter in the `nudr-drservice` section is set to `WHITELISTED`, EIR response in EIC as follows:

```

{
  "status": "WHITELISTED"
}

```

When you perform a GET request with PEI and IMEI together with Subscription Permanent Identifier (SUPI) and Generic Public Subscription Identifier (GPSI), if the PEI and SUPI or PEI and GPSI matches are not found in the EIR database, then EIR responds as per the existing `defaultResponse` configuration present in `nudr-drservice` section:

```

{ "title": "Not Found",
  "status": 404,
  "detail": "Provided SUPI/GPSI does not match with PEI association",
  "cause": "ERROR_EQUIPMENT_UNKNOWN",
  "invalidParams": []
}

```

To Observe

The following metrics are introduced:

- `eir_default_response_imei_not_found`

See UDR [Metrics](#) section for more information on metrics.

2.23.2 International Mobile Subscriber Identity (IMSI) Fallback Lookup Enhancement for EIR

Earlier in EIR, if the incoming message had International Mobile Equipment Identity (IMEI), International Mobile Subscriber Identity (IMSI), and Mobile Station Integrated Services Digital Network (MSISDN) and if only the IMEI was provisioned in the EIR database without IMSI or MSISDN, then EIR used to respond with a configured default response for the user equipment status. Using IMSI Fallback Lookup Enhancement feature, EIR returns user equipment status as WHITELISTED, BLACKLISTED, or GREYLISTED for the matched IMEI in the EIR database.

The feature is enabled using the `imsiLookupFallbackEnabled` feature flag. The default value is set as `false`.

The IMSI Fallback Lookup Enhancement feature enables EIR to perform the following functionalities:

Note

The incoming request message must contain the Subscription Permanent Identifier (SUPI)/Generic Public Subscription Identifier (GPSI) and Permanent Equipment Identifier (PEI)/IMEI.

- Search for PEI and IMSI received in the request message with the PEI and IMSI in the EIR database. If matched, EIR responds with the associated equipment status. The equipment status values are WHITELISTED, BLACKLISTED, or GREYLISTED.
- Search for only PEI match in the EIR database. If matched, EIR responds with the associated equipment status.
- Search for PEI match in the EIR database with any other IMSIs. If found, EIR responds as per the default configuration in `defaultResponse` parameter. The default values are EQUIPMENT_UNKNOWN, WHITELISTED, BLACKLISTED, or GREYLISTED.
- If PEI match is not found in the EIR database, then EIR responds as per the default configuration in `defaultResponseIMEINotFound` parameter. The default values are EQUIPMENT_UNKNOWN, WHITELISTED, BLACKLISTED, or GREYLISTED.

Managing IMSI Fallback Lookup Enhancement

Enable

You can enable the IMSI Fallback Lookup Enhancement feature using Helm Configuration:

- You can enable or disable the feature by setting the `imsiLookupFallbackEnabled` parameter to `true` or `false` in the global section of the `ocudr-custom-values.yaml` file. The default value is set to `false`. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

```
# EIR Configurations
eirConfigurations :
  defaultResponse: "EQUIPMENT_UNKNOWN"
  accessLogEnabled: "true"
```

```
defaultResponseIMEINotFound: "EQUIPMENT_UNKNOWN"
imsiLookupFallbackEnabled: "false"
```

Configure

You can configure this feature using REST API and CNC Console.

- **Configure using CNC Console:** You can enable or disable the feature by setting the `imsiLookupFallbackEnabled` field to true or false in the [Global Configurations](#) in the CNC Console.
- **Configure using REST API:** You can perform the PUT operation on the `http://10.75.229.45:30015/nudrconfig/v1/udr.global.cfg/GLOBAL` to enable or disable the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "dbServiceName": "mysql-connectivity-service.occne-rs1",
  "udrServices": "n5g-eir-eic",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "sbiCorrelationInfoEnable": false,
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "version": "v3",
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "vsaDefaultBillingDay": 1,
  "subscriberActivityEnabled": false,
  "addDefaultBillingDay": true,
  "enableControlledShutdown": false,
  "keyType": "msisdn",
  "keyRange": "000000-000000",
  "dbConflictResolutionEnabled": false,
  "maxNumberOfSubscriptions": 30,
  "suppressNotificationEnabled": true,
  "configClientConnectTimeout": 1000,
  "configClientReadTimeout": 1000,
  "s13InterfaceEnabled": false,
  "defaultResponse": "EQUIPMENT_UNKNOWN",
  "accessLogEnabled": true,
  "defaultResponseIMEINotFound": "EQUIPMENT_UNKNOWN",
  "imsiLookupFallbackEnabled": true,
  "subscriberIdentifiers": {
    "msisdn": [],
    "imsi": [],
    "nai": [],
    "extid": []
  }
}
```

Observe

There are no metrics, alerts, or KPIs for this feature.

Upgrade and Rollback of IMSI Fallback Lookup Enhancement Feature

- **Upgrade scenario:** When you upgrade UDR from previous release version by setting `imsiLookupFallbackEnabled` parameter as true, then the feature is enabled in the upgraded release version.
- **Rollback scenario:** When you rollback UDR to previous release version the configuration is also rolled back and the feature does not work.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.23.3 Diameter S13 Interface

Equipment Identity Register (EIR) is a database containing records of all User Equipment (UE) that are allowed or disallowed from accessing the network. Each UE is identified by its International Mobile Equipment Identity (IMEI). When an UE is identified by the network, the Mobility Management Entity (MME) requests the IMEI of the UE, which is sent to the EIR for authorization. This feature supports diameter interface between EIR and MME to retrieve the UE equipment status of the subscriber.

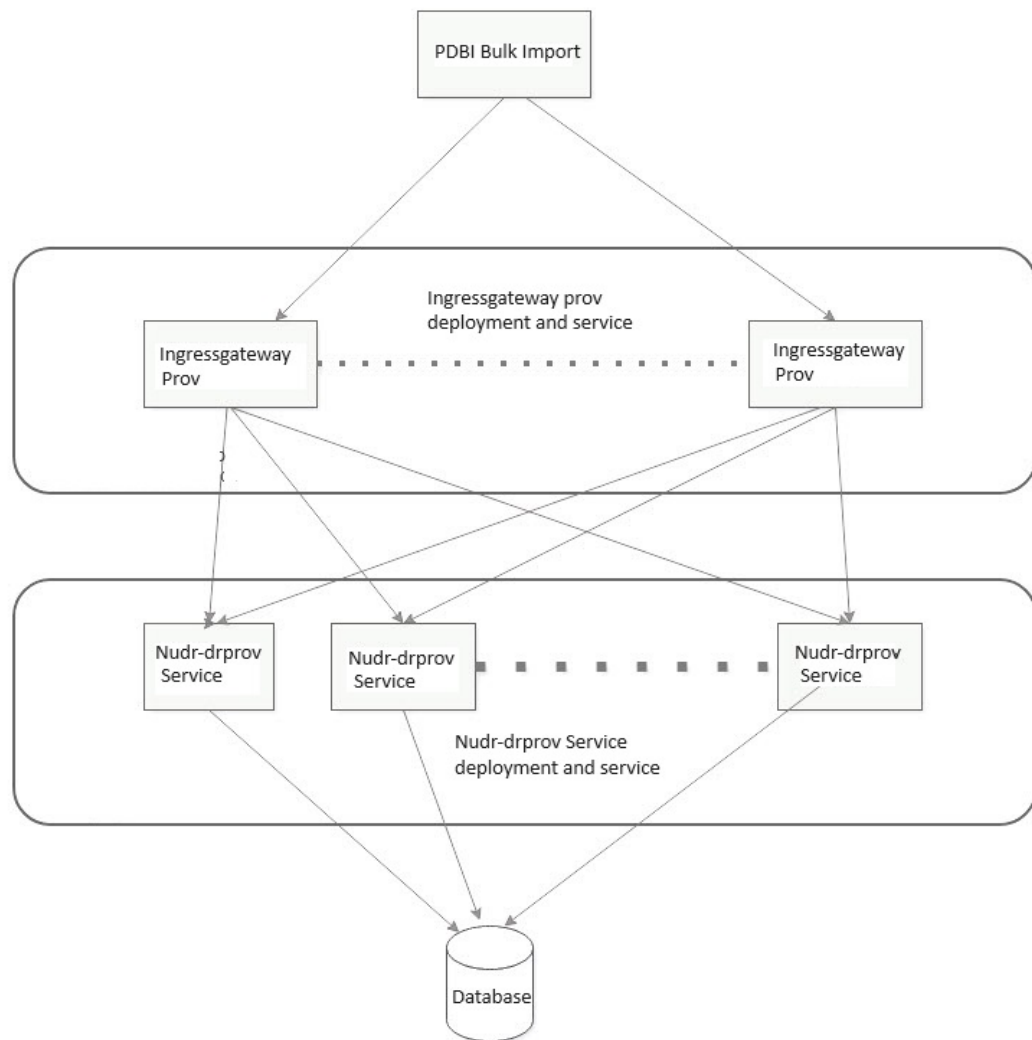
The following microservices are introduced in EIR mode to support Diameter S13 Interface:

- **nudr-diam-gateway:** It is used to establish and maintain the diameter connection.
- **nudr-diameterproxy:** It is used to retrieve the equipment status from the EIR database, which is provisioned using provisioning system or PDBI subscriber bulk import tool.

Equipment Data Provisioning

The equipment data is provisioned in EIR using the PDBI subscriber bulk import tool or provisioning API. For more information, see [Subscriber Bulk Import Tool](#) and [Support for 5G EIR Signaling](#).

The bulk import PDBI interface sends the request to the Ingress Gateway provisioning API with the subscriber equipment data and then the request is forwarded to the **nudr-drprov service**, which sends the equipment data to the database.

Figure 2-22 Equipment Data Provisioning

Diameter Gateway Support

Diameter Gateway microservice is used in EIR to get the equipment status of the subscriber using diameter S13 interface. Diameter Gateway supports the following functionalities:

- Establishing and maintaining the diameter connection with the WHITELISTED peer
- Horizontal pod auto scaling and High Availability (HA) for the traffic load distribution
- Georedundancy deployment models
- Diameter Gateway Pod Congestion feature, which protects the pod by rejecting the traffic when congested. For more information about the feature, see [Diameter Gateway Pod Congestion Control](#).

For the diameter peer to send the EC request to retrieve the equipment status of the subscriber, you must perform the following configuration on Diameter Gateway microservice:

- Diameter peer must be WHITELISTED to establish the connection with EC. Configure the diameter peer details under `allowedClientNodes` in the diameter gateway configurations.

- Perform the PUT operation on the `http://10.75.229.85:31430/nudr-config/v1/udr.diamgateway.cfg/DIAMETER-GATEWAY` to whitelist the diameter peer. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "realm": "oracle.com",
  "identity": "nudr.oracle.com",
  "reconnectDelay": 3,
  "responseTimeout": 4,
  "connectionTimeOut": 3,
  "watchdogInterval": 6,
  "reconnectLimit": 50,
  "onDemandMigrationEnabled": false,
  "allowedClientNodes": [{
    "port": 3868,
    "identity": "peer1",
    "host": "10.75.185.161",
    "responseOnly": false,
    "realm": "seagull",
    "type": "server"
  }]
}
```

Mobile Equipment Identity Check Request (ECR)

When provisioning of the equipment data is done using PDBI import or provisioning API, the equipment status is retrieved using ECR request as follows:

- A connection is established from the diameter peer using the Capability Exchange Request (CER) request. The Capability Exchange Answer (CEA) responds to the request and maintains the connection.
- ECR request is sent from diameter peer to diameter gateway. The request is then sent to diameter proxy from diameter gateway to get the equipment status from the EIR database.
- Equipment status of the subscriber is returned to the diameter peer.

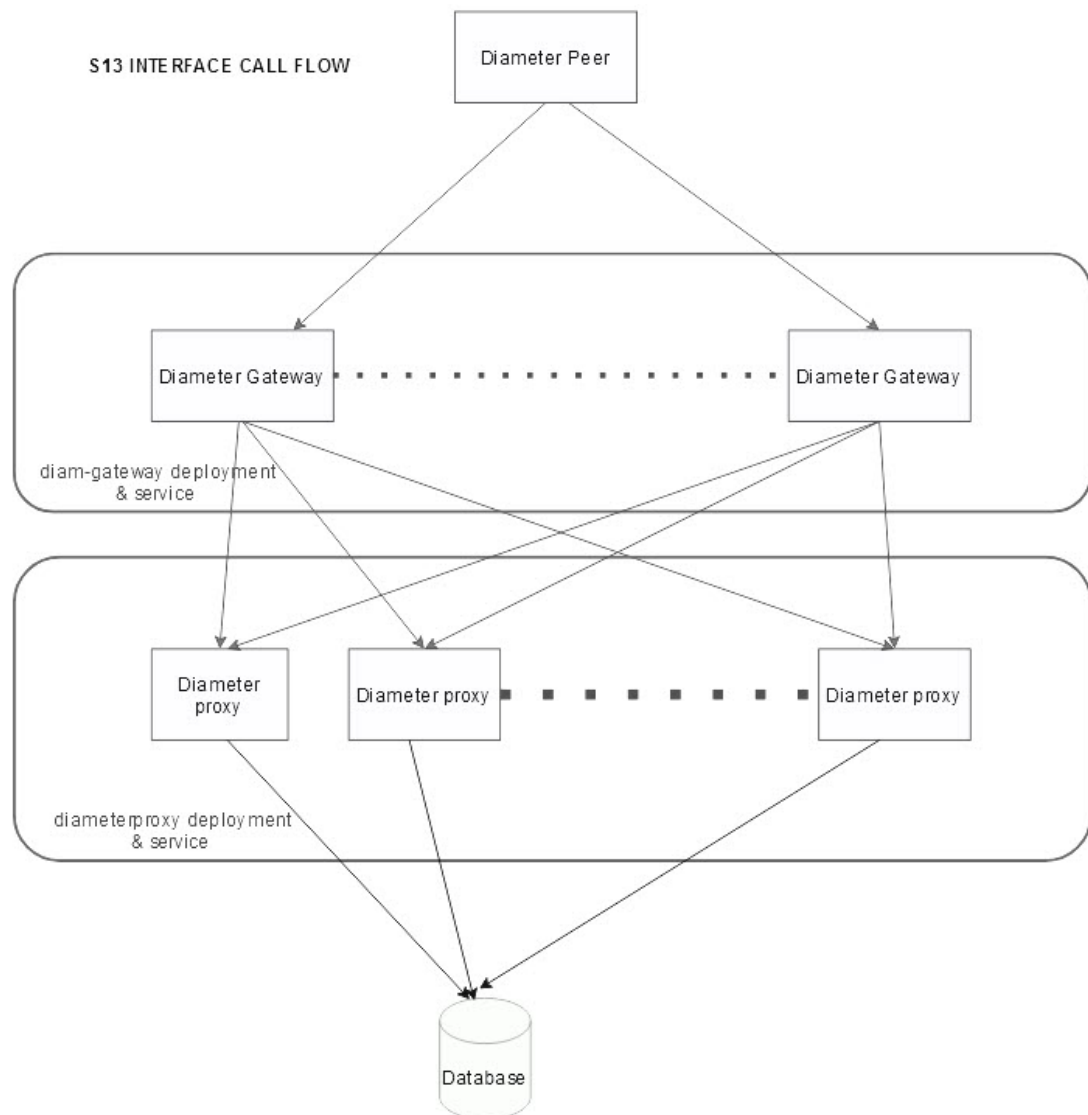
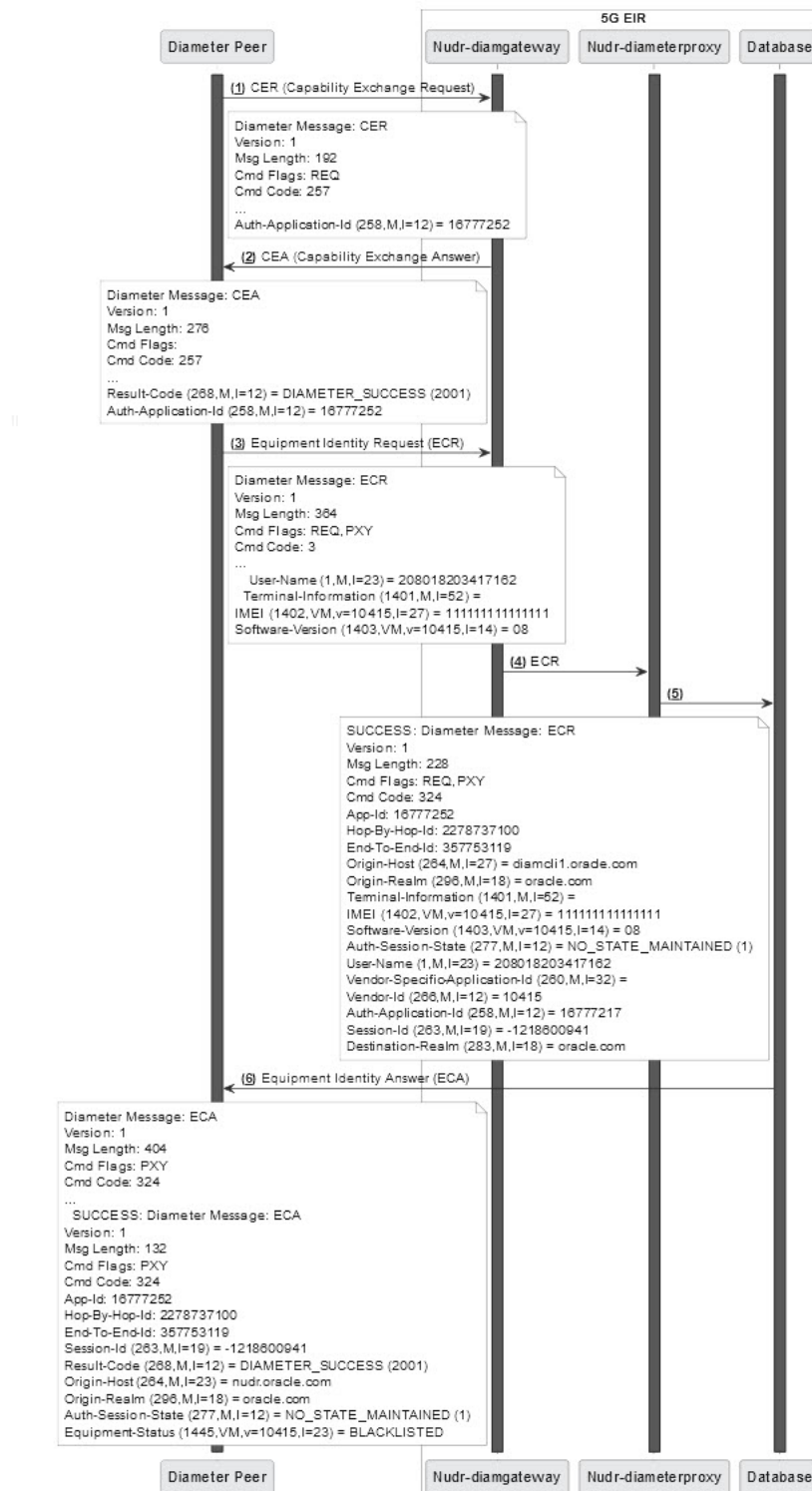
Figure 2-23 Diameter S13 Interface Call Flow

Figure 2-24 ECR Request



Equipment Status Retrieval Call Flow

The following steps details the process the equipment status retrieval based on the ECR request received:

- If ECR contains software version as part of terminal information along with IMEI, lookup will be based on IMEI and software version, if the software version is not available, then look up is based on IMEI only.
- If IMEI or IMEI software version is not found in the EIR database, then ECA returns with configured defaultResponseIMEINotFound.
- If IMEI or IMEI software version is found in EIR database, then you must check if user name is present in the ECR request. If user name is not present in the ECR request, then ECA returns with DIAMETER SUCCESS and equipment status.
- If user name is present, then you must check if a match is present in the EIR database.
- If user name match is not present with IMEI or IMEISV, then you must send ECA with configured defaultResponse.
- If user name match is present with IMEI or IMEI software version, then ECA returns with DIAMETER SUCCESS and equipment status.

Figure 2-25 Equipment Status Retrieval Call Flow

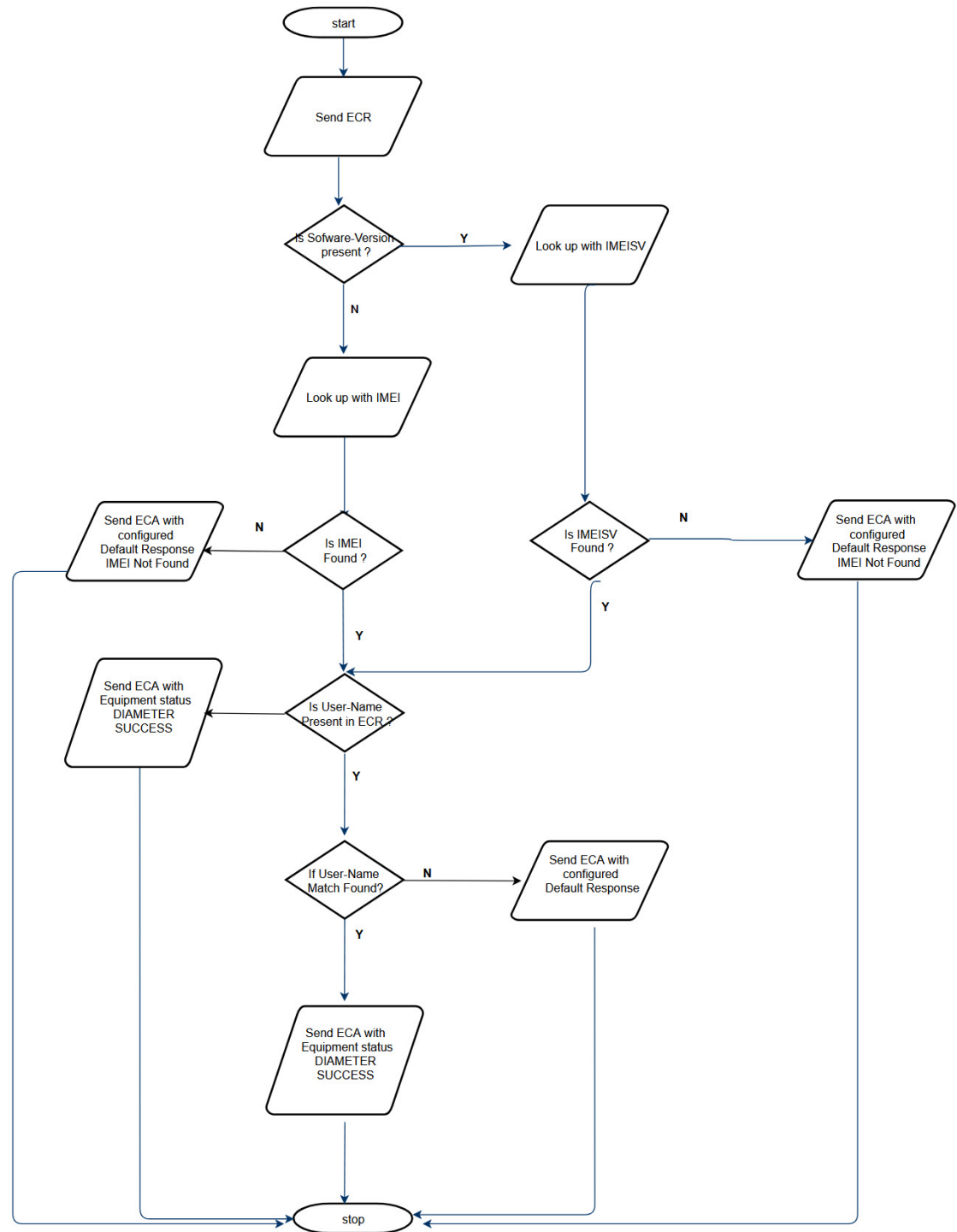


Table 2-32 Possible Error Codes

Result Code	Details
2001	Error Response: DIAMETER_SUCCESS Error Message: diameter success

Table 2-32 (Cont.) Possible Error Codes

Result Code	Details
5422	Error Response: DIAMETER_ERROR_EQUIPMENT_UNKNOWN Error Message: unknown equipment
5005	Error Response: DIAMETER_MISSING_AVP Error Message: missing avp
5004	Error Response: DIAMETER_INVALID_AVP_VALUE Error Message: invalid avp value
5001	Error Response: DIAMETER_AVP_UNSUPPORTED Error Message: diameter avp not supported
5009	Error Response: DIAMETER_AVP_OCCUR_TOO_MANY_TIMES Error Message: diameter avp occurs too many times
5010	Error Response: DIAMETER_NO_COMMON_APPLICATION Error Message: no common application
3004	Error Response: DIAMETER_TOO_BUSY Error Message: diameter is busy

EIR Access Log and Default Response

This feature supports EIR access logs and default response implementation. For more information, see [Support for 5G EIR Signaling](#).

Managing Diameter S13 Interface

Enable

You can enable the Diameter S13 Interface feature using Helm configuration:

Note

The existing EIR configurations `defaultResponse`, `defaultResponseIMEINotFound`, and `accessLogEnabled` parameters are applicable for this feature.

- You can enable or disable the feature by setting the `s13InterfaceEnable` parameter to `true` or `false` in the global section of the `ocudr-custom-values.yaml` file. The default value is set to `false`. For more information on this parameter, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

```
s13InterfaceEnable: true
# EIR Configurations
eirConfigurations:
  defaultResponse: "EQUIPMENT_UNKNOWN"
  accessLogEnabled: "true"
  defaultResponseIMEINotFound: "EQUIPMENT_UNKNOWN"
```

Configure

You can configure this feature using REST API and CNC Console.

Configure using CNC Console

You can enable or disable the feature by setting the `s13InterfaceEnable` field to `true` or `false` in the [Global Configurations](#) in the CNC Console.

Configure using REST API

You can perform the PUT operation on the `http://10.75.229.45:30015/nudr-config/v1/udr.global.cfg/GLOBAL` to enable or disable the feature. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```

{
  "dbServiceName": "mysql-connectivity-service.occne-ndb",
  "udrServices": "All",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "sbiCorrelationInfoEnable": false,
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceeb-44bb-95e0-f6a55a328b03",
  "vsaDefaultBillingDay": 1,
  "subscriberActivityEnabled": false,
  "addDefaultBillingDay": true,
  "enableControlledShutdown": true,
  "keyType": "msisdn",
  "keyRange": "000000-000000",
  "nfType": "UDR",
  "nfFqdn": "udr001.oracle.com",
  "suppressNotificationEnabled": true,
  "configClientConnectTimeout": 1000,
  "configClientReadTimeout": 1000,
  "s13InterfaceEnable": true,
  "defaultResponse": "EQUIPMENT_UNKNOWN",
  "accessLogEnabled": true,
  "defaultResponseIMEINotFound": "EQUIPMENT_UNKNOWN",
  "imsiLookupFallbackEnabled": false,
  "subscriberIdentifiers": {
    "msisdn": [],
    "imsi": [],
    "nai": [],
    "extid": []
  }
}

```

Observe

Metrics

The following metrics are used for this feature:

- eir_default_response_imei_not_found
- eir_txn_default_response_sent

Alerts:

The following alerts are used for this feature:

- NudrDiameterProxyServiceDown
- NudrDiameterGatewayDown
- OceirDiameterCongestionDocState
- OceirDiameterCongestionCongestedState
- DiameterPeerConnectionsDropped

Upgrade and Rollback of Diameter S13 Interface Feature

During upgrade from 24.2.x release to 24.3.0, all the existing configurations under nudr-drservice configuration such as defaultResponse, defaultResponseIMEINotFound, and accessLogEnabled are present under the global configuration. During rollback from 24.3.0 release version to previous release versions the configurations are restored to nudr-drservice configuration.

Note

Parameters s13InterfaceEnabled and imsiLookupFallbackEnabled are added to the global configurations.

Maintain

If you encounter alerts at system or application levels, see the [Alert Details](#) section for resolution steps.

In case the alerts still persist, perform the following:

1. Collect the logs: For more information on collecting logs, see *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

2.24 Support for LCI and OCI Header

This feature introduces Load Control Information (LCI) and Overload Control Information (OCI) Headers in the HTTP request(s) and response(s). These headers send the operating status of the producer resources to the consumer Network Functions (NFs).

The load information in the headers include the following:

- LCI Header: This header sends the current load information of the producer Network Function (NF). This information helps the consumer NF in load balancing the traffic in the

network. 3gpp-Sbi-Lci denotes LCI Header. When the conditions for generating LCI Header are met, the NF service producer includes this header at the NF scope, back-end service scope, or both.

- **OCI Header:** This header sends the current overload information so that the peer NF can reduce the traffic towards the overloaded NF. `3gpp-Sbi-Oci` denotes OCI Header. When the conditions for generating OCI Header are met, the NF service producer includes this header at the NF scope, back-end service scope, or both.

Following are the examples of LCI and OCI Headers:

Note

For LCI and OCI headers, reporting of the load is considered based on `localLciHeaderValidity` and `loadThreshold`. Once the `localLciHeaderValidity` expires the `loadThreshold` value will not be considered. Threshold value will be validated only within the validity period.

```
3gpp-Sbi-Lci: Timestamp: "Tue, 07 Jun 2021 01:11:22 GMT"; Load-Metric: 25;
slf-FQDN: slf.example.com
```

```
3gpp-Sbi-Oci: Timestamp: "Tue, 04 Feb 2020 08:49:37 GMT"; Period-of-Validity:
120s; Overload-Reduction-Metric: 75; SLF-FQDN: slf.example.com
```

Modify the `ocudr-custom-values.yaml` file with the following configuration parameters:

LCI and OCI Header Priorities

You must extract the consumer NF identity from one of the request headers mentioned below at `ingressgateway-sig`:

- OAuth token (1st priority)
- User-Agent (2nd priority)
- Via (3rd priority)

If there are multiple headers available at ingressgateway-sig, then the consumer NF identity must be extracted based on the priority order of the header as mentioned above.

The NF Instance ID is extracted from the above headers and this Instance ID act as the required key to identify a consumer NF. Example given below:

oAuth token header

Authorization → NF instance ID is extracted after decoding the below token
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiI2ZmFmMWJiYy02ZTRhLTQ0NTQtYTUwNy1hMTRlZjZhbmWJjMTEiLCJzdWIiOiJmZTdkOTkyYi0wNTQxLXRjN2Q2YWI4NC1jNmQzMGIxYXJAcXJlLCJhdWQiOiJTTUYiLCJzY29wZSI6Im5zbWYtcGRlc2Vzc2lvbiIsImV4cCI6MTY0NDQ5MjE5OTh0.xGud7dIW EugyXV-I8M-xld90Ar_rKBB3y_4_SWVVAwHyb6DxtWUZ5uB-KtXwozbrJorMNkF6rwz5D0G6VA m9WLw

User agent header

```
User-Agent → NFType-NFInstanceId fqdn
UDR-6faf1bbc-6e4a-4454-a507-a14ef8e1bc11 abc.com
```

Via header

Via → HttpVersion fqdn
 2.0 abc.5gc.mnc012.mcc311.pub.3gppnetwork.org
 (Note - Since NF-Instance ID is not a part of Via header, fqdn will be used as the required key)

Configuration

Ingress Gateway

Enable

You can enable LCI and OCI Headers globally by setting the `lciHeaderConfig.enabled` and `ociHeaderConfig.enabled` parameters as `true`, respectively.

Configure

You can configure LCI and OCI Headers using the Helm based configuration:

Sample Helm configuration:

```
lciHeaderConfig:
  enabled: true
  loadThreshold: 30
  localLciHeaderValidity: 60000 #(value in milliseconds)
  producerSvcIdHeader: "UDRHeader"

ociHeaderConfig:
  enabled: true
  producerSvcIdHeader: "UDRHeader"
  validityPeriod: 30000 #(value in milliseconds)
  overloadConfigRange: #Note - minor, major and critical conditions should
  cover complete range of 0 to 100 both inclusive for it to be a valid config
    minor: "[75-80]"
    major: "[81-90]"
    critical: "[91-100]"
  reductionMetrics:
    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)

nfInstanceId: *nfInsId #Producer NF Instance ID
svcToSvcInstanceIdMapping:
- svcName: "ocudr-nudr-drservice"
  serviceInstanceId: *nfInsId

perfInfoConfig:
  pollingInterval: 5000 #(value in milliseconds)
  serviceName: "nudr-perf-info"
  host: nudr-perf-info
  port: 5905
  perfInfoRequestMap: "/load"
```

Note

During configuring the minor, major, or critical range numbers make sure that the range numbers specified do not overlap each other. For example 75-80 and 80-85. This can give you incorrect result.

Egress Gateway**Enable**

You can enable LCI and OCI Headers globally by setting the `lciHeaderConfig.enabled` and `ociHeaderConfig.enabled` parameters as `true`, respectively.

Configure

You can configure LCI and OCI Headers using the Helm based configuration:

Sample Helm configuration:

```
lciHeaderConfig:
  enabled: true
  loadThreshold: 40
  localLciHeaderValidity: 1000 #(value in milliseconds)
  consumerSvcIdHeader: "ConsumerHeader"
  producerSvcIdHeader: "UDRHeader"

ociHeaderConfig:
  enabled: true
  consumerSvcIdHeader: "ConsumerHeader"
  producerSvcIdHeader: "UDRHeader"
  validityPeriod: 5000 #(value in milliseconds)
  overloadConfigRange: #Note - minor, major and critical conditions should
  cover complete range of 0 to 100 both inclusive for it to be a valid config
    minor: "[75-80]"
    major: "[81-90]"
    critical: "[91-100]"
  reductionMetrics:
    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)

nfInstanceId: *nfInsId #Producer NF Instance ID
svcToSvcInstanceIdMapping:
- svcName: "ocudr-nudr-drservice"
  serviceInstanceId: *nfInsId

perfInfoConfig:
  pollingInterval: 5000 #(value in milliseconds)
  serviceName: "nudr-perf-info"
  host: nudr-perf-info
  port: 5905
  perfInfoRequestMap: "/load"
```

Note

During configuring the minor, major, or critical range numbers make sure that the range numbers specified do not overlap each other. For example 75-80 and 80-85. This can give you incorrect result.

2.25 Support for User-Agent Header

User Agent Header Validation in Ingress Gateway

User-Agent header helps the producer Network Function (NF) to identify the consumer NF that has sent the request. To implement this, 3GPP introduced the use of User-Agent header for consumers to include the same in service requests. Additionally, producers may require to support the validation of the User-Agent headers to complete the request identification process in the network.

This feature validates the User-Agent header in the incoming request against the configured values in Ingress Gateway.

The following format is used to generate User-Agent header:

<NFType>-<NFInstanceID> <NF FQDN>
where, <NFType> is the type of the Network Function.

<NFInstanceID> is the instance ID of the NF.

<NF FQDN> is the FQDN of the NF.

Note

You must provide the correct format as follows for this feature to work, providing only the NFTType without the hyphen will lead to request failure. User-Agent headers must be in one of the following sample formats:

- PCF-6c03761c-fc42-47a5-9c37-cb8c520fb56f
- PCF-
- PCF-6c03761c-fc42-47a5-9c37-cb8c520fb56f pcf.oracle.com

Below screenshot shows how the User-Agent header is passed in the request:

[illegible]

Following validations are made once the feature is enabled:

- If the user-agent header is present, then it is matched with the configured NF types. If a match is found, then validation is successful and request is allowed to pass. If a match is not found, then request is rejected with a configurable error code.

- If the user-agent header is present and has multiple values, then the request is rejected with a configurable error code. Hence, the user-agent header, if present should have a single value.
- If the user-agent header is not present and validationType is relaxed, then validation is not made and request is allowed to pass. If validationType is strict, then request will be rejected with a configurable error code.
- If the configured NF types are not PCF and NRF, then the configuration is rejected in the REST mode. In case of Helm mode, the startup probe fails.

Managing User-Agent Header

Enable

Perform the following configurations to enable the user-agent header feature:

- **Helm:** Set the value for property userAgentHeaderValidationConfigMode to Helm and parameter userAgentHeaderValidation to true in the custom-values.yaml file under the Ingress Gateway section. For more information, see the section Server Header at Ingress Gateway under Chapter Customizing Cloud Native Core Policy in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.
- **REST API:** Set the value for property userAgentHeaderValidationConfigMode to REST, and send a PUT request, along with payload by setting flag userAgentHeaderValidation to true.

Note

By default, this feature is disabled with userAgentHeaderValidationConfigMode as **REST**.

For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Configure

You can configure this feature either through HELM or REST.

The configuration applies to all the routes. By default, feature is disabled and mode of configuration is REST. Scope of configuration remains same throughout the deployment lifecycle. That is, If configMode is HELM, then only HELM based configuration is supported throughout the deployment. REST based configuration is not supported in between.

- **Configure using Helm:**

```
#User-Agent header validator configuration
#Mode of configuration. Can be either HELM or REST
userAgentHeaderValidationConfigMode: REST
userAgentHeaderValidation:
  enabled: false
  # If User-Agent header is not present or it's value is null in the
  incoming request then validation type can be used to skip or perform
  validation. If set to strict then validation will be performed.
  # If set to relaxed then validation will be skipped.
  validationType: strict
  # List of consumer NF Types to be matched against the value of User-
```

```
Agent header in the request
consumerNfTypes:
- "PCF"
- "NRF"
```

- **Configure using REST API:** You can perform the PUT operation on the `http://10.75.229.75:31187/udr/nf-common-component/v1/igw-sig/useragentheadervalidation` REST API to configure the feature.

```
{
  "enabled": false,
  "validationType": "strict",
  "consumerNfTypes": [
    "PCF", "NRF"
  ]
}
```

For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Observe

See [UDR Metrics](#) and [Alert Details](#) sections for more information on metrics and configurable parameters, respectively.

User Agent Header Generation in Egress Gateway

This feature generates the User-Agent header to help the producer NF identify the consumer NF that sent the request.

The generated User-Agent header is in the following format:

<NF Type>-<Instance-Id> <FQDN>, only if `addFqdnToHeader` is set to true.
Otherwise, the generated header is in the following format:

<NF Type>-<Instance-Id>
where, <NF Type> is the type of the Network Function.

<Instance-Id> is the instance ID of the NF.

<FQDN> is the FQDN of the NF.

Example:

```
'User-Agent': 'UDR-6c03761c-fc42-47a5-9c37-cb8c520fb56f udr.oracle.com'
#If addFqdnToHeader is set to true
```

```
'User-Agent': 'UDR-6c03761c-fc42-47a5-9c37-cb8c520fb56f'
#If addFqdnToHeader is set to false
```

Managing User-Agent Header

Enable

Perform the following configurations to enable the user-agent header feature:

- **Helm:** Set the value for property `userAgentHeaderConfigMode` to `Helm` and parameters `userAgentHeader` and `overwriteHeader` to `true` in the `custom-values.yaml` file under the `Egress Gateway` section.
- **REST API:** Set the value for property `userAgentHeaderConfigMode` to `REST`, and send a `PUT` request, along with payload by setting `enabled` and `overwriteHeader` to `true`.

Note

By default, this feature is disabled with `userAgentHeaderValidationConfigMode` as **REST**. For `egressgateway`, to generate the User Agent header using the `<NF Type>` and `<Instance-Id>` provided in the custom values, the `overwriteHeader` flag should always be set to `true`. If the `addFqdnToHeader` flag is set to `true`, `egressgateway` adds `fqdn` to the header only if the `<FQDN>` value is provided in the payload.

For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Configure

You can configure this feature either through HELM or REST.

The configuration applies to all the routes. By default, feature is disabled and mode of configuration is `REST`. Scope of configuration remains same throughout the deployment lifecycle. That is, If `configMode` is `HELM`, then only `HELM` based configuration is supported throughout the deployment. `REST` based configuration is not supported in between.

- **Configure using Helm:**

```
#User-Agent header generation configuration
userAgentHeaderConfigMode: REST
userAgentHeader:
  enabled: false # flag to enable or disable the feature
  nfType: "" # NF type of consumer NF
  nfInstanceId: "" # NF type of consumer NF
  addFqdnToHeader: false # Flag to add fqdn. If enabled then user-agent
header will be generated along with
                        the fqdn configured otherwise fqdn will not be
added
  nfFqdn: "" #fqdn of NF. This is not the fqdn of gateway
  overwriteHeader: false

oauthClient:
  enabled: false
  nfInstanceId: fe7d992b-0541-4c7d-ab84-c6d70b1b01b1
  nfType: PCF
```

- **Configure using REST API:** You can perform the `PUT` operation on the `http://10.75.229.75:31187/udr/nf-common-component/v1/egw/useragentheader` REST API to configure the feature.

```
"useragentheader": {
  "enabled": false,
  "nfType": " UDR",
  "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1",
```

```
"nfFqdn": "udr.oracle.com",  
"addFqdnToHeader": false,  
"overwriteHeader": true  
}
```

For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

2.26 Diameter Gateway Pod Congestion Control

The Diameter Gateway is a diameter proxy agent for UDR. Being a front-end microservice for diameter traffic for both Ingress Gateway and Egress Gateway, the Diameter Gateway can get congested due to higher traffic, higher CPU usage, and higher memory utilization. Thus, it is imperative to have suitable congestion control features in place for Diameter Gateway pods to avoid adverse impacts on latency and performance. Another reason for the need for a congestion control mechanism for Diameter Gateway is the nature of diameter connections. These connections are long lived and are distributed by external LoadBalancer. When LoadBalancer routes an incoming request from network to Diameter Gateway pod, it does not take into account health or load of the pod. As a result, uneven distribution of traffic may happen and gateway pods experience congestion.

In UDR, a congestion control mechanism has been implemented that allows the system to perform the following tasks:

- Determine the pod congestion state
- Trigger Congestion Control

Determining Pod Congestion State

At any given point in time, a pod can be in any one of the following states:

1. Normal
2. DOC (Danger of Congestion)
3. Congested

Each state is calculated based on the threshold values configured for the following resources:

- **CPU:** CPU usage of the container is monitored and congestion state is calculated based on the following parameter - `cpuacct.usage` that provides current CPU usage in nano seconds.

Triggering Congestion Control

Every time a message is fetched from the ring buffer for processing, the system checks the current congestion state of the pod. If the current state is either DOC or Congested, the congestion control mechanism is invoked. After verifying that the message type is a request, a priority is assigned to it. If the assigned priority is less than or equal to discard priority, the message is rejected.

Note

Congestion control does not apply to response messages as they are always accepted.

The priority value for a request can be from 0 to 15 where 0 is the highest priority and 15 is the lowest priority.

Enable

To enable the feature:

In the Diam-gateway section of the custom-values.yaml file, set the value of the **podCongestion.enabled** flag to true. By default, it is false.

Configure

You can customize the following configurations related to this feature using the CNC Console or REST APIs for UDR:

- Default priority configuration for all the messages.
- Sample count threshold for congestion state change.
- Diameter Request message priority configuration when pod in DOC and CONGESTED state.
- DOC and CONGESTED state configuration.
- Threshold percentage for Cpu, Memory and Queue usage for DOC and CONGESTED state.
- **Configure using CNC Console:** Perform the feature configurations on the Diameter Gateway Congestion Configurations page. For more information about the configurations, see [Diameter Gateway Congestion](#).
- **Configure using REST API:** UDR provides the REST API for configuring diameter gateway pod congestion control feature. You can perform the PUT operation to configure the feature. For more information about REST API configuration, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

The default configuration details for Diameter Pod Congestion Control feature are:

Deployment resource

- 1 Diameter Gateway pod with default resource (2cpu, 2 Gb memory)
- 3 diameterproxy pods with default resource (2cpu, 2Gb memory)

and Diameter traffic model

- 33% Subscription with getting profile
- 33% update the profile data
- 33% get profile data

Table 2-33 Congestion Configuration

Pod state	CPU Usage
Normal	< 60%
DOC (Danger of Congestion)	60% to 80%
Congested	> 80%

Note

Initial pod state set to NORMAL and state will change based on sample count and one of the resource usage goes more than configured number.

Observe

UDR uses the pod congestion metrics for this feature. For more information, see the section [Diameter Gateway Metrics](#). Alerts are raised when the following metric is pegged:

- ocudr_pod_congestion_state

Note

Prometheus automatically injects name of the pod with label name "kubernetes_pod_name" to metric. This information is further used for alerting purposes.

For more information about alerts, see [Alert Details](#).

Maintain

Error logs are generated when the system is congested and the actions taken to bring the system back to normal. Warning logs are generated to indicate the congestion level. However, error logs are not generated when messages are rejected to avoid additional resource usage to write error logs.

The following is a sample error log:

```
{
  "instant": {
    "epochSecond": 1644819199, "nanoOfSecond": 415860635, "thread": "pool-7-
    thread-1", "level": "WARN", "loggerName": "ocpm.cne.common.congestion.CongestionSt
    ateManager", "message": "Abrupt
      change in congestion state, current state : NORMAL, calculated state :
    CONGESTED,
      transientStateSampleCount :

0", "endOfBatch": false, "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger", "thr
    eadId": 44, "threadPriority": 5, "messageTimestamp": "2022-02-14T06:13:19.415+0000"
    , "application": "ocudr", "engVersion": "1.15.50", "mktgVersion": "1.15.0.0.0", "micr
    oservice": "diam-gateway", "vendor": "Oracle"}
  {"instant": {
    "epochSecond": 1644819200, "nanoOfSecond": 722676387, "thread": "pool-7-
    thread-1", "level": "WARN", "loggerName": "ocpm.cne.common.congestion.CongestionSt
    ateManager", "message": "Current
      Congestion state changed, old state : DOC, new state : NORMAL,
    transientStateSampleCount : 10

", "endOfBatch": false, "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger", "thre
    adId": 44, "threadPriority": 5, "messageTimestamp": "2022-02-14T06:13:20.722+0000"
    , "application": "ocudr", "engVersion": "1.15.50", "mktgVersion": "1.15.0.0.0", "micro
    service": "diam-gateway", "vendor": "Oracle"}
```

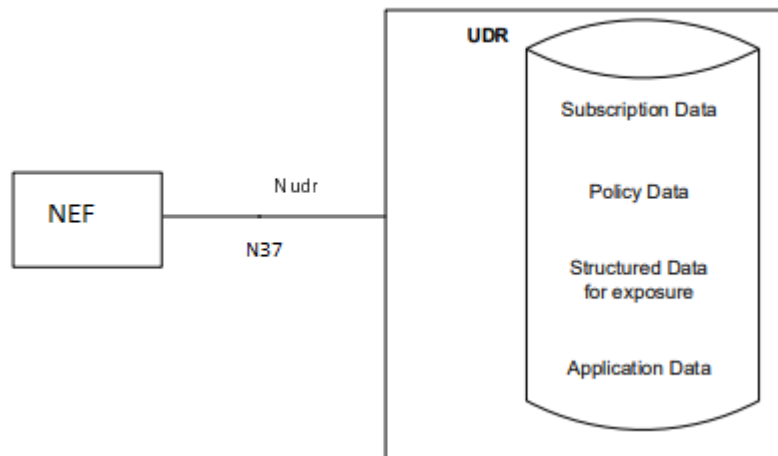
2.27 Support for Network Exposure Function API

This feature allows:

- UDR to support exposure data for signaling (Access and Mobility data and PduSessionManagementData) and NEF Subscription and Notification APIs.
- Provisioning of the exposure data into the UDR by enhancing the existing provisioning interface.

NEF uses Nudr interface to access the structured data for exposure stored in the UDR. Here, the UDR is acting as NF Service Producer. It provides Unified Data Repository service to the NF service consumer, NEF.

Figure 2-26 NEF Integration



This feature supports the following service operations through Nudr_DataRepository for the exposure data:

- Query: The Query service operation is used for the NEF to retrieve the data stored in the UDR.
- Create: The Create service operation is used for the NEF to create the data into the UDR.
- Update: The Update service operation is used for the NEF to update the data stored in the UDR.
- Delete: The Delete service operation is used for the NEF to remove the data from the UDR.
- Subscribe: The Subscribe service operation is used for the NEF to subscribe to the data change notification from the UDR.
- Unsubscribe: The Unsubscribe service operation is used for the NEF to unsubscribe to the data change notification from the UDR.
- Notify: The Notify service operation is used for the UDR to notify the data change notification to the NEF.

Note

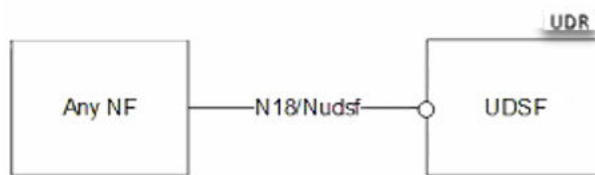
For more information, see 3GPP specification 29.519.

For more information on operations supported for NEF data, see the **Operations Supported for NEF Data** section in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

2.28 Supports Unstructured Data Storage Function

UDR supports **Unstructured Data Storage Function (UDSF)** NF functionality. It allows other 5G NFs to store and retrieve unstructured data. Its specifications are not defined by 3GPP and its part of Oracle's 5G UDR solution.

Figure 2-27 Relation between Any NF and UDSF



For more information on UDSF Data Operations, see the **UDSF Data Operations** section in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Note

For structured data types, see 3GPP specification 29.504 and 29.519

2.29 Support for Default Group ID in SLF

Overview

When a subscriber is not provisioned in the Subscriber Location Function (SLF), the SLF responds with an error stating "Subscriber does not exist". This feature provides an option of not responding with an error but instead provides a default group or default NF for the subscribers that are not provisioned in the SLF.

Based on the default group `nfgroupId` configured we can control the NFType for which default group ID is applicable. If the feature is disabled, then the response for non provisioned subscribers will be "Subscriber does not exist". Default `slfGroupName` gets inserted during installation or upgrade to `SLF_GROUP_NAME` table using helm hooks. You can update the Default `slfGroupName` using the SLF GroupName Configuration API. For information about SLF GroupName Configuration API, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Note

If subscriber is not provisioned in the SLF and default ID group feature is enabled, then SLF look up will give a response as 200 OK, provided that the default group is configured with nfGroupID including the nfType requested in the query parameter.

To Enable

To enable this feature using CNC Console or Helm chart changes:

- **Enable Using CNC Console:** To enable this feature using CNC Console, set the **defaultGroupIDEnabled** field as true in the **Data Repository Service Configurations** page under **Service Configurations**.
- **Enable Using values.yaml file:** To enable this feature, set the **defaultGroupIDEnabled** flag as 'true' under the **nudr-drservice** section of default SLFGroupName.

To Configure default SLFGroupName

To configure default SLFGroupName using CNC Console or helm chart changes:

- **Configure Using CNC Console:** To configure default SLFGroupName using CNC Console application, set the **defaultSLFGroupName** field as DefaultGrp.
- **Configure Using REST API:** To configure default SLFGroupName using the values.yaml file, set the value of the **defaultSLFGroupName** parameter in the file.

For information about these parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

To Observe

See [UDR Metrics](#) and [UDR KPIs](#) sections for more information on metrics, alerts, and KPIs respectively.

2.30 Subscriber Location Function

Overview

In large deployments, network operators may choose to deploy an SLF, which helps in identifying a subscriber to the particular Network Function(NF). This allows the network operators to support ranges based on subscriber identities for NF's.

You can configure UDR to support the storage and retrieval of this information by acting as a Subscriber Location Function (SLF). This SLF information is consumed by NRF. This 5G SLF solution supports Nudr-groupid-map service as defined by 3GPP and registers with NRF for this service. UDR supports REST or JSON based provisioning APIs for SLF data.

Note

To understand SLF Lookup Support on UDR, it is recommended to see the SLF Lookup APIs as defined in the 3GPP 29.504 v16.2.0

To Configure

To configure SLF over UDR network using CNC Console or REST API:

- **Configure Using CNC Console:** To configure SLF using CNC Console application, see [Provisioning - SLF Data](#).
- **Configure Using REST API:** To configure SLF using REST APIs, see the following sections in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.
 - Operations Supported for SLF Data
 - Provisioning Operations for SLF Data

To Observe

See [UDR Metrics](#) and [UDR KPIs](#) sections for more information on metrics, configurable parameters, and KPIs respectively.

2.30.1 Error Response Enhancement for SLF Lookup Request

SLF supports the “cause” attribute in the problem details for all error responses as per 3GPP 29.504 specification. The “cause” attribute in the problem details object of the HTTP error response payload is used to indicate the application-related information for that specific occurrence of the error. This enables the HTTP client to take appropriate action. The error response is enhanced to support error causes scenario for SLF lookup requests.

The following table lists the error response code for SLF.

Table 2-34 Error Response Code

Protocol or Application Error	Description	Scenario
USER_NOT_FOUND	When the subscriber UE ID received in the NF Group ID retrieval request does not exist in the database. HTTP Status Code: 404 NOT Found	This error response is caused due to the following scenario: • A signaling request when subscriber corresponding to UE ID or NF group does not exist in the database. • A provisioning request to read or delete a subscriber when the subscriber does not exist in the database.
GROUP_IDENTIFIER_NOT_FOUND	When the subscriber UE ID retrieval request is received in the NF Group ID and the subscriber exists in the database but the NF Group ID is not configured for the NF type requested. HTTP Status Code: 404 NOT Found	This error response is caused due to the following scenario: • A signaling request when subscriber exist in database but the requested NF type is not configured for that particular group. • A provisioning request is sent to delete a particular group which does not exist in database. • A provisioning request is sent to get details of a particular group which does not exist in database. • A provisioning request sent to provision a subscriber with a group that does not exist in database.

Table 2-34 (Cont.) Error Response Code

Protocol or Application Error	Description	Scenario
MANDATORY_QUERY_PARAM_MISSING	Query parameter which is defined as mandatory or conditional but which is mandatory requirement for an HTTP method is not included in the URI of the request. HTTP Status Code: 400 BAD REQUEST	This error response is caused due to the following scenario: A request where a required query parameter for that API is missing.
MANDATORY_QUERY_PARAM_MISSING	A mandatory query parameter or conditional query parameter but which is mandatory requirement for an HTTP method that was received in the URI with semantically incorrect value. HTTP Status Code: 400 BAD Request	This error response is caused due to the following scenario: - A request where the value of query parameter is incorrect. For example, the following request is sent with incorrect subscriber id: nudr-group-id-map/v1/nf-group-ids?subscriber-id=msisd-1111111111&nf-type=UDM nudr-group-id-map/v1/nf-group-ids?subscriber-id=msisd1111111111&nf-type=UDM - If the value of the query parameter is blank or subscriber id length is less than three digits.
INVALID_QUERY_PARAM	The HTTP request contains an unsupported query parameter in the URI. HTTP Status Code: 400 BAD REQUEST	This error response is caused due to the following scenario: A request which has an invalid query parameter. For example: The following request is sent with incorrect subscriber id value: nudr-group-id-map/v1/nf-group-ids?subscriber-id=msisdn-11111111111&nf-type=UDM&ABC=DEF
MANDATORY_IE_INCORRECT	A mandatory IE (which can be within the JSON body, a variable part of an apiSpecificResourceUriPart, or an HTTP header) or conditional IE but mandatory requirement for an HTTP method that was received with a semantically incorrect value. HTTP Status Code: 400 BAD REQUEST	This error response is caused due to the following scenario: - A request with an invalid JSON. - Entry for URI (API_NAME/ API_VERSION/ BASE_RESOURCE_NAME) combination does not exist in the resource map.

Table 2-34 (Cont.) Error Response Code

Protocol or Application Error	Description	Scenario
OPTIONAL_IE_INCORRECT	An optional Information Element (IE) (which can be within the JSON body or an HTTP header) for an HTTP method that was received with a semantically incorrect value which prevents successful processing of the service request. HTTP Status Code: 400 BAD Request	This error response is caused due to the following scenario: • Schema validation errors. • A request which does not contain key value as that of the URI. • A request which does not contain a known key.
SUBSCRIBER_LINKED_TO_GROUP	When deleting a group which has subscribers linked to that particular group. HTTP Status Code: 403 FORBIDDEN	This error response is caused due to the following scenario: • A request to delete a group which has subscribers linked.

2.31 Server Header Inclusion in Error Response

The Network Function interacting with UDR can determine the originator of HTTP error response only if the originator of an error includes a Server header. You can add the Server header attribute in the outgoing error response of UDR or SLF with NF name and NF instance id registered with NRF.

Format: Server: <NF_Name_registered_in_NRF>-<NFInstanceID_Registered_In_NRF>

Example: Server: UDR-5a7bd676-ceed-44bb-95e0-f6a55a328b03

You can enable and configure the server header feature using Helm configuration and REST API.

Configure using HELM

You can enable the server header by setting the `serverHeaderDetails: enabled` parameter to `true` in the `ocudr-custom-values.yaml` file. This includes all UDR or SLF 4xx and 5xx responses indicating details of the server header, where the problem originated. To change the default configuration, modify the `serverHeaderDetails` and `errorCodeSeriesList` values in the `ocudr-custom-values.yaml` file. For more information about the `ocudr-ingressgateway` microservice parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

Configure using REST API

You can perform the GET, PUT, and PATCH operations to configure the server header feature.

1. Configure the `errorcodeserieslist` on `http://<NUDR_CONFIG_SERVICE_IP>:<NUDR_CONFIG_PORT>/udr/nf-common-component/v1/igw-sig/errorcodeserieslist`.

```
[
  {
    "id": "E1",
    "errorCodeSeries": [
```



```

        {
            "errorSet": "4xx",
            "errorCodes": [
                -1
            ]
        },
        {
            "errorSet": "5xx",
            "errorCodes": [
                -1
            ]
        }
    ],
    },
    {
        "id": "E2",
        "errorCodeSeries": [
            {
                "errorSet": "4xx",
                "errorCodes": [
                    -1
                ]
            }
        ]
    }
]

```

2. Configure serverheaderdetails on `http://<NUDR_CONFIG_SERVICE_IP>:<NUDR_CONFIG_PORT>/udr/nf-common-component/v1/igw-sig/serverheaderdetails`

Note

- The `errorCodeSeriesId` is a mandatory attribute. The value must be one among the `errorCodeSeriesList` resource defined in `/udr/nf-common-component/v1/igw-sig/errorcodeserieslist` endpoint.
- The `nfType` is a mandatory attribute. Server header will not be included in the response, if `nfType` is not defined.
- The `nfInstanceId` is an optional attribute. Only `nfType` is used for server header value, if `nfInstanceId` is not defined. You must provide the same value used in `customs values.yaml` file for `nfInstanceId`.

```

{
    "enabled": true,
    "errorCodeSeriesId": "E1",
    "configuration": {
        "nfType": "UDR",
        "nfInstanceId": "5a7bd676-ceed-44bb-95e0-f6a55a328b03"
    }
}

```

To Maintain

In case of failure or misconfiguration, server header does not populate in UDR or SLF responses. In this case, it is recommended to follow the instructions given in the [Configure using HELM](#) and [Configure using REST API](#) sections.

2.32 Rate Limiting

Rate Limiting means setting the number of HTTP requests that UDR can receive in a given period. With the implementation of this feature, UDR can restrict the traffic for a particular route.

The UDR user is required to send the **sbi-priority** header in the request. Based on the value of the sbi-priority header, the requests are discarded. If the request does not contain this header, then this feature is not invoked and the request is not discarded.

Note

This feature supports only priority based discard.

When a request contains the **sbi-priority** header:

- All the routes configured under the routesconfiguration REST API and its corresponding rate limit configuration details are screened.
- Starts calculating the number of requests that should be dropped in each sampling period specific to the incoming request route + HTTP method.
- At the end of each sampling period, rate calculator notifies the route level, rate limit about the rate calculated per route+{Method}.
- Requests with **sbi-priority** header with a value greater than the configured number are discarded. Requests with lesser values are rejected based on configured error code.
- The number of messages dropped in the current sampling period are decided based on the extra number of messages received than configured limit in previous sampling period.

To Enable

By default, UDR loads all the rate limiting configuration payloads while installing or upgrading UDR when Ingress Gateway default configuration is used. Once the UDR installation or upgrade completes, the user can edit the routelevelratelimiting.json payload to send a PUT request to `http://10.75.229.75:31187/udr/nf-common-component/v1/igw/routelevelratelimiting`.

To enable the rate limiting feature, set the ratelimit parameter to 'true' in the routelevelratelimiting.json payload.

To Configure

To configure the rate limiting feature on UDR, the user is required to do the REST based configurations on Ingress Gateway, which are stored in the configuration database of UDR.

For more information about REST based APIs, see the **Configuration APIs for Rate Limit** section in the **Configuration API Specification** chapter of the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

2.33 Overload Handling

Overload occurs when 100% of the planned capacity is exhausted. It can be due to uneven distribution of traffic towards a given UDR service instance, network fluctuations leading to traffic bursts, or unexpected high traffic volume at any given point of time.

During overload conditions, the service's response time may grow to unacceptable levels, and exhaustion of resources may force service to unexpected behavior or even downtime. Overload management helps to control services downtime and ensures service availability during extreme overload conditions.

Using Ingress Gateway and perf-info microservices, UDR provides options to control network congestion and handle the overload scenarios.

REST API's support for Overload Control Feature

The below REST API's are supported for **ocpolicymapping**:

- GET: /udr/nf-common-component/v1/igw-sig/ocpolicymapping
- PUT: /udr/nf-common-component/v1/igw-sig/ocpolicymapping

```
{
  "enabled": true,
  "mappings": [
    {
      "svcName": "testDeploy-nudr-drservice",
      "policyName": "OCDP1"
    }
  ],
  "samplingPeriod": 6000
}
```

Table 2-35 Request/Response Parameters

Field Name	Data type	Description
enabled	boolean	Set the values to true or false to enable or disable the overload control feature.
mappings.svcName	string	The SVC entry determine mapping between services and discards policy name per service.
mappings.policyName	string	The discard policy entry determine mapping between services and discards policy name per service.
samplingPeriod	integer	Time frame for each cycle of overload control per service. The value is in milliseconds.

The below REST API's are supported for **ocdiscardpolicies**:

- GET: /udr/nf-common-component/v1/igw-sig/ocdiscardpolicies

- PUT: /udr/nf-common-component/v1/igw-sig/ocdiscardpolicies

```
[
  {
    "name": "OCDP1",
    "scheme": "PercentageBased",
    "policies": [
      {
        "level": "L1",
        "value": 0,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L2",
        "value": 5,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L3",
        "value": 15,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L4",
        "value": 25,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error503"
      }
    ]
  },
  {
    "name": "OCDP2",
    "scheme": "PriorityBased",
    "policies": [
      {
        "level": "L1",
        "value": 30,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L2",
        "value": 24,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L3",
        "value": 15,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      }
    ]
  }
]
```

```

        "level": "L4",
        "value": 10,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error503"
    }
]

```

Table 2-36 Request/Response Parameters

Field Name	Data type	Description
name	string	Name of the discarded policy.
scheme	string	Discarded policy scheme.
policies.value	integer	Value of priority, above which requests are considered as potential candidates for drop. It is the percentage of requests to drop in the current sampling period over the calculated rate in the previous sampling period.
policies.action	string	Defines the action to be taken on selected request rejections based on the error code.
policies.level	string	Defines the overload level.
policies.errorCodeProfile	string	The error code profiles.

The below REST API's are supported for **errorcodeprofiles**:

- GET: /udr/nf-common-component/v1/igw-sig/errorcodeprofiles
- PUT: /udr/nf-common-component/v1/igw-sig/errorcodeprofiles

```

[
  {
    "name": "error429",
    "errorCode": 429,
    "errorCause": "Too many reequets",
    "errorTitle": "Too many reequets",
    "redirectURL": "",
    "retry-after": "",
    "errorDescription": "Too many reequets"
  },
  {
    "name": "error503",
    "errorCode": 503,
    "errorCause": "Backend not able to handle traffic",
    "errorTitle": "Backend not able to handle traffic",
    "redirectURL": "",
    "retry-after": "",
    "errorDescription": "Backend not able to handle traffic"
  }
]

```

Table 2-37 Request/Response Parameters

Field Name	Data type	Mandatory (M) or Optional (O)	Description
name	string	M	Name of the error.
errorCode	string	M	The errorCode field in an errorScenario which determines the HttpStatusCode that need to be populated in the ProblemDetails (HttpStatus field) response from ingress gateway, when the exception occurred at ingress gateway matches the configured errorScenario exceptionType field.
errorCause	string	O	The errorCause field in an errorScenario determines the error cause that need to be populated in the ProblemDetails (cause field) response from ingress gateway when the exception occurred at ingress gateway matches the configured errorScenario exceptionType parameter.
errorTitle	string	O	The errorTitle field in an errorScenario determines the title that needs to be populated in ProblemDetails (Title field) response from ingress gateway, when the exception occurred at ingress gateway matches the configured errorScenario's exceptionType parameter.

Table 2-37 (Cont.) Request/Response Parameters

Field Name	Data type	Mandatory (M) or Optional (O)	Description
redirectURL	string	O	The redirectUrl field in an errorScenario determines the redirection URL, this value is populated in location header when sending the response from ingress gateway. The header is populated only when the exception occurred at ingress gateway matches the configured errorScenario's exceptionType parameter. The errorCode configured for the particular errorScenario lies in 3xx error series and the redirectUrl field for the particular errorScenario is configured appropriately.
retry-after	string	O	The retryAfter field in an errorScenario determines the value in seconds/particular date after which the service must be retried. This value is populated inRetry-After header when sending the response from ingress gateway. The header is populated only when the exception occurred at ingress gateway matches the configured errorScenario's exceptionType parameter. The errorCode configured for the particular errorScenario lies in 3xx error series and the retryAfter field for the particular errorScenario is configured appropriately in seconds.

Table 2-37 (Cont.) Request/Response Parameters

Field Name	Data type	Mandatory (M) or Optional (O)	Description
errorDescription	string	O	The errorDescription field in an errorScenario determines the description that need to be populated in ProblemDetails (Detail field) response from ingress gateway when the exception occurred at ingress gateway matches the configured errorScenario's exceptionType field.

The below REST API's are supported for **overloadthresholdconfigurations**:

- GET: /udr/nf-common-component/v1/perf-info/overloadLevelThreshold
- PUT: /udr/nf-common-component/v1/perf-info/overloadLevelThreshold

```
[
  {
    "svcName": "ocudr-nudr-drservice",
    "metricsThresholdList": [
      {
        "metricsName": "svc_failure_count",
        "levelThresholdList": [
          {
            "level": "L1",
            "onsetValue": 13000,
            "abatementValue": 12500
          },
          {
            "level": "L2",
            "onsetValue": 14000,
            "abatementValue": 13500
          },
          {
            "level": "L3",
            "onsetValue": 15000,
            "abatementValue": 14500
          },
          {
            "level": "L4",
            "onsetValue": 16000,
            "abatementValue": 15500
          }
        ]
      }
    ],
    {
      "metricsName": "memory",
      "levelThresholdList": [
```



```

        {
            "level": "L1",
            "onsetValue": 50,
            "abatementValue": 45
        },
        {
            "level": "L2",
            "onsetValue": 70,
            "abatementValue": 65
        },
        {
            "level": "L3",
            "onsetValue": 85,
            "abatementValue": 82
        },
        {
            "level": "L4",
            "onsetValue": 90,
            "abatementValue": 88
        }
    ]
},
{
    "metricsName": "cpu",
    "levelThresholdList": [
        {
            "level": "L1",
            "onsetValue": 65,
            "abatementValue": 60
        },
        {
            "level": "L2",
            "onsetValue": 75,
            "abatementValue": 70
        },
        {
            "level": "L3",
            "onsetValue": 80,
            "abatementValue": 75
        },
        {
            "level": "L4",
            "onsetValue": 90,
            "abatementValue": 85
        }
    ]
},
{
    "metricsName": "svc_pending_count",
    "levelThresholdList": [
        {
            "level": "L1",
            "onsetValue": 13000,
            "abatementValue": 12500
        },
        {

```

```

        "level": "L2",
        "onsetValue": 14000,
        "abatementValue": 13500
    },
    {
        "level": "L3",
        "onsetValue": 15000,
        "abatementValue": 14500
    },
    {
        "level": "L4",
        "onsetValue": 16000,
        "abatementValue": 15500
    }
  ]
}

```

Table 2-38 Request/Response Parameters

Field Name	Data type	Description
svcName	string	Name of the back end service (svcName).
metricsThresholdList	array	List of criteria used to calculate the load level.
metricsThresholdList.metricsName	string	Name of the criteria
metricsThresholdList.levelThresholdList	array	List of threshold values
metricsThresholdList.levelThresholdList.level	string	Specifies the name of the level. The name specified in this parameter must match with the level name in the Ingress Gateway's ocdiscardpolicies.
metricsThresholdList.levelThresholdList.onsetValue	string	If the metric level goes below this value, load level goes to a lower value. - This value should be greater than 0. - This value should be greater than onset value.
metricsThresholdList.levelThresholdList.abatementValue	string	If the metric level goes above this value, load level is set.

To Enable Overload Control

To enable the overload control feature through:

- custom-values.yaml file:
 - In the global section of the custom-values.yaml file, enable the **overloadManagerEnabled** flag.
 - In the perf-info section of the custom-values.yaml file, perform the following changes:
 - * envMysqlDatabase: Provide the configdb name of the deployed setup.

The **configdbName** is the value, which is assigned while creating Kubernetes secret at the time of UDR installation.

- * prometheus: Provide the prometheus url of the deployed setup. For Example : 10.75.229.87:31403/cne-2230-rc2/prometheus/.
- Using REST API endpoint:
 - To enable the ocpolicymapping API:
 - * Send the GET request to `http://<nudr-configsvcname:port>/udr/nf-common-component/v1/igw-sig/ocpolicymapping`.

```
{
  "enabled": false,
  "mappings": [
    {
      "svcName": "<ReleaseName>-nudr-drservice",
      "policyName": "OCDP1"
    }
  ],
  "samplingPeriod": 6000
}
```

- * Edit the following parameters in the response payload:
 - * Set the enabled flag as 'true' in the ocpolicymapping json file and
 - * Change svcName to <ReleaseName>. For example, if svcName is testDeploy, you must update as testDeployin the below payload.
- * Send the PUT request to `http://<nudr-configsvcname:port>/udr/nf-common-component/v1/igw-sig/ocpolicymapping`

```
{
  "enabled": true,
  "mappings": [
    {
      "svcName": "testDeploy-nudr-drservice",
      "policyName": "OCDP1"
    }
  ],
  "samplingPeriod": 6000
}
```

- To validate ocdiscardpolicy API:
 - * Send the GET request to `http://<nudr-configsvcname:port>/udr/nf-common-component/v1/igw-sig/ocdiscardpolicies`

```
[
  {
    "name": "OCDP1",
    "scheme": "PercentageBased",
    "policies": [
      {
        "level": "L1",
        "value": 0,
        "action": "RejectWithErrorCode",

```

```

        "errorCodeProfile": "error429"
      },
      {
        "level": "L2",
        "value": 5,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L3",
        "value": 15,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L4",
        "value": 25,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error503"
      }
    ]
  },
  {
    "name": "OCDP2",
    "scheme": "PriorityBased",
    "policies": [
      {
        "level": "L1",
        "value": 30,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L2",
        "value": 24,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L3",
        "value": 15,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error429"
      },
      {
        "level": "L4",
        "value": 10,
        "action": "RejectWithErrorCode",
        "errorCodeProfile": "error503"
      }
    ]
  }
]

```

By default the percentages are configured for **OCDP1**.

- * To change the percentage, send PUT request along with above discard policy payload `http://<nudr-configsvcname:port>/udr/nf-common-component/v1/igw-sig/ocdiscardpolicies` .

Note

You should not use memory as a overload threshold configuration for overload control feature because the memory accumulated by the pods during the traffic run is not released by the pods even when the traffic is reduced. This causes the service to remain in high overload levels causing the traffic to be discarded.

Set the `overloadthresholdconfigurations` API to control the overload levels. By default below configurations are present when you deploy UDR:

- * Send the GET request to `http://<nudr-configsvcname:port>/udr/nf-common-component/v1/perf-info/overloadLevelThreshold`.

```
[
  {
    "svcName": "ocudr-nudr-drservice",
    "metricsThresholdList": [
      {
        "metricsName": "svc_failure_count",
        "levelThresholdList": [
          {
            "level": "L1",
            "onsetValue": 13000,
            "abatementValue": 12500
          },
          {
            "level": "L2",
            "onsetValue": 14000,
            "abatementValue": 13500
          },
          {
            "level": "L3",
            "onsetValue": 15000,
            "abatementValue": 14500
          },
          {
            "level": "L4",
            "onsetValue": 16000,
            "abatementValue": 15500
          }
        ]
      },
      {
        "metricsName": "memory",
        "levelThresholdList": [
          {
            "level": "L1",
            "onsetValue": 50,
            "abatementValue": 45
          }
        ]
      }
    ]
  }
]
```

```

        {
            "level": "L2",
            "onsetValue": 70,
            "abatementValue": 65
        },
        {
            "level": "L3",
            "onsetValue": 85,
            "abatementValue": 82
        },
        {
            "level": "L4",
            "onsetValue": 90,
            "abatementValue": 88
        }
    ]
},
{
    "metricsName": "cpu",
    "levelThresholdList": [
        {
            "level": "L1",
            "onsetValue": 65,
            "abatementValue": 60
        },
        {
            "level": "L2",
            "onsetValue": 75,
            "abatementValue": 70
        },
        {
            "level": "L3",
            "onsetValue": 80,
            "abatementValue": 75
        },
        {
            "level": "L4",
            "onsetValue": 90,
            "abatementValue": 85
        }
    ]
},
{
    "metricsName": "svc_pending_count",
    "levelThresholdList": [
        {
            "level": "L1",
            "onsetValue": 13000,
            "abatementValue": 12500
        },
        {
            "level": "L2",
            "onsetValue": 14000,
            "abatementValue": 13500
        },
    ]
}

```

```

        "level": "L3",
        "onsetValue": 15000,
        "abatementValue": 14500
      },
      {
        "level": "L4",
        "onsetValue": 16000,
        "abatementValue": 15500
      }
    ]
  }
}

```

To update the default values of CPU, pending, and failure count, send **PUT** request with modified values of payload.

When overload control feature is enabled, it is enabled for the percentage based discard policy of UDR. To enable overload control feature for priority based discard policy (optional):

- Send the GET request to `http://ocudr-ingress-gateway:80/udr/nf-common-component/v1/igw-sig/ocpolicymapping`.
- Change the `policyName` to 'OCDP2' from OCDP1. OCDP1 denotes the percentage based discard policy.
- Send the PUT request to `http://ocudr-ingress-gateway:80/udr/nf-common-component/v1/igw-sig/ocpolicymapping`.

This enables the overload control feature for priority based discard policy.

For more information about the configurable parameters related to overload handling, see the **Configuring Unified Data Repository** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

To Validate

The overload control feature can be validated based on the following threshold configurations:

- If traffic reaches L1 level, traffic will not be discarded and alarm will be raised
- If traffic reaches L2 level , 5 percentage traffic will be discarded and alarm will be raised
- If traffic reaches L3 level ,15 percentage traffic will be discarded and alarm will be raised
- If traffic reaches L4 level , 25 percentage traffic will be discarded and alarm will be raised

If priority based discard policy is enabled, then UDR identifies requests with `sbi-priority` defined and discards all the requests that has header value greater than 30. When the CPU reaches the 65-70 percentage, it is on L1.

If percentage based discard policy is enabled, then discard percentage is 0 and no traffic is discarded.

The default configuration details for overload threshold levels are:

- L1Alert name: `OcudrOverload60Percent`

Note

You should not use memory as a overload threshold configuration for overload control feature because the memory accumulated by the pods during the traffic run is not released by the pods even when the traffic is reduced. This causes the service to remain in high overload levels causing the traffic to be discarded.

Table 2-39 Default Configuration Details for Overload Threshold Levels

Type	L1Alert name: OcudrOverload60Percent			
-	Onset	Abatement	Discard %	Discard Priority
CPU	65	60	0	30
Memory	50	45	0	30
Pending Count	13000	12500	0	30
Failure Count	13000	12500	0	30

- L2 Alert name:OcudrOverload75Percent

Table 2-40 Default Configuration Details for Overload Threshold Levels

Type	L2 Alert name:OcudrOverload75Percent			
-	Onset	Abatement	Discard %	Discard Priority
CPU	75	70	5	24
Memory	70	65	5	24
Pending Count	14000	13500	0	30
Failure Count	14000	13500	0	30

Note

The default error code for a discard is 429 for all the types defined above.

- L3 Alert name:OcudrOverload80Percent

Table 2-41 Default Configuration Details for Overload Threshold Levels

Type	L3 Alert name:OcudrOverload80Percent			
-	Onset	Abatement	Discard %	Discard Priority
CPU	80	75	15	15
Memory	85	82	15	15
Pending Count	15000	14500	15	15
Failure Count	15000	14500	15	15

Note

The default error code for a discard is 429 for all the types defined above.

- L4 Alert name:OcudrOverload90Percent

Table 2-42 Default Configuration Details for Overload Threshold Levels

Type	L4 Alert name:OcudrOverload90Percent			
-	Onset	Abatement	Discard %	Discard Priority
CPU	90	85	25	10
Memory	90	88	25	10
Pending Count	16000	15500	25	10
Failure Count	16000	15500	25	10

Note

The default error code for discard is 503 for all the types defined above.

Note

If there is any breach at multiple threshold levels, then the highest threshold level is considered and appropriate action is taken. UDR also sends an alert for appropriate levels.

To configure this feature, you can use REST based API configurations. For more information, see the **Configuration APIs for Overload Handling** section of the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

To Observe

If there is a breach at any threshold level, then UDR sends an alert. For more information about alerts, see [Alert Details](#).

Overload Control to Support Provisioning Traffic

To support overload control for provisioning traffic, additional perf-info-prov service and ingressgateway-prov service are needed.

Ingressgateway-prov fetches the load values of the provisioning traffic from perf-info-prov.

The following configurations are required to support overload control for provisioning traffic:

- custom-values.yaml file
 - Enable the `perf-info-prov.enabled` flag. By default, it is false.
- REST
 - Modify all the configuration Rest URLs related to the Ingress Gateway overload configurations to `igw-prov` in place of `igw-sig` in the URL .
 - Modify all the configuration Rest URLs related to the `perf-info` overload configurations to `perf-info-prov` in place of `perf-info` in the URL.
 - Modify all the payloads `svcName` to `ocudr-nudr-dr-provservice` in place of `ocudr-nudr-dr-service`.

Metrics

When the overload control feature is enabled, the below metrics are available.

Table 2-43 Overload Control Feature Metrics

Metric Name	Metric Description	Metric Type	Labels
load_level	Tracks load_level of a service.	Gauge	Service = <ReleaseName>-nudr-drservice
service_resource_stress	Tracks CPU, failures, and pending requests based on which the service calculates the overload level.	Gauge	type = "PendingRequest","CPU","Memory","FailureCount" Service = <ReleaseName>-nudr-drservice
service_resource_overload_level	Tracks individual resource overload level.	Gauge	type = "PendingRequest","CPU","Memory","FailureCount" Service = <ReleaseName>-nudr-drservice

2.34 Oauth2.0

Overview

Oauth 2.0 is a security feature that authenticate requests received from valid consumer NFs. The consumer NF requests for access token from the issuer, 'NRF' and uses this access token to send the request to UDR.

UDR authenticates the request received and either processes or discards it based on access token validation received in request. The access token is validated with the configured public key certificate in UDR. To validate access token, ensure UDR is configured with:

- NF Type and NF Instance ID of UDR
- Services hosted at UDR
- NRF public Key certificate

For UDR deployment, the provisioning traffic bypass Oauth2.0 validation by default using the route level configuration setting.

Note

By default, the Oauth2.0 feature is not enabled for Ingress Gateway Provisioning.

Oauth behavior is based on validationType. There are two types of validation as follows:

- Strict validation type: If Oauth token is received in the request header, then Ingress Gateway validates the token. If the request do not have the Oauth token, then the request fails with missing token.
- Relaxed validation type: If Oauth token is received in the request header, then Ingress Gateway validates the token. If the request do not have the Oauth token, then the request is processed further ignoring the validation.

To Enable and Configure

To enable Oauth2.0 validation, configure both Helm based and REST based configurations on Ingress Gateway.

Helm based Configuration

For Helm based configuration, you need to:

- Enable the `oauthValidatorEnabled` parameter on Ingress Gateway
- Create a secret that stores NRF public key certificate. To create a secret, run the following command:

```
kubectl create secret generic oauthsecret --from-file=public_key.crt -n ocudr
```

In the above command:

- `oauthsecret` is the secret name
- `ocudr` is the namespace
- `public_key.crt` is the public key certificate

Note

You can have any number of certificates in the secret.

- Configure the secret and namespace on Ingress Gateway in the OAUTH CONFIGURATION section of the `values.yaml` using following fields:

```
nrfPublicKeyKubeSecret: oauthsecret  
nrfPublicKeyKubeNamespace: ocudr
```

After Helm configuration, send the REST requests to use configured public key certificates.

REST based Configuration

Using REST based configuration, you can distinguish between the certificates configured on different NRFs and can use these certificates to validate the token received from a specific NRF.

For REST based configuration, use CNCC GUI.

Note

The CNCC GUI uses config API and payload as follows:
 /{nfType}/nf-common-component/v1/{serviceName}/{resource}" with service name as "igw" and resource as "oauthvalidatorconfiguration".

Payload:

```
{
    "keyIdList": [{
        "keyId":
"664b344e74294c8fa5d2e7dfaaba407",
        "kSecretName": "samplesecret1",
        "certName": "samplecert1.crt",
        "certAlgorithm": "ES256"
    }],
    "instanceIdList": [{
        "instanceId":
"664b344e74294c8fa5d2e7dfaaba407",
        "kSecretName": "samplesecret2",
        "certName": "samplecert2.crt",
        "certAlgorithm": "ES256"
    }],
    "oAuthValidationMode": "INSTANCEID_ONLY"
}
```

Using CNCC GUI, you can configure multiple keyId and instanceId objects of different NRFs. Based on NRF instanceId received in token, UDR validates the token. It allows you to configure instanceId of NRF or keyId configured in NRF, secret name in which the certificate is stored, certificate name and certificate algorithm. After the REST request is sent, Oauth validator considers a certificate from the secret configured.

Using oAuthValidationMode, you can select the mode of validation as either INSTANCEID_ONLY, KID_ONLY or KID_PREFERRED. Based on the mode of validation selected, it checks keyIdList or instanceIdList. The KID_PREFERRED is a fall back mode where it checks for keyId in the token. If keyId is present in the token and matches with the configured keyId, then the validation mode is KID_ONLY otherwise it is INSTANCEID_ONLY.

Oauth validation should not be enabled for specific type of requests such as all provisioning requests sent to UDR. In the routesConfig section of custom-values.yaml file, set the oauth validator flag as 'false'.

Figure 2-28 oauthValidator not enabled

```
- id: traffic_mapping_http_group_prov
  uri: http://{ .Release.Name }-nudr-drservice:5001
  path: /nudr-group-id-map-prov/**
  order: 6
  metadata:
    oauthValidator:
      enabled: false
```

Oauth Error Codes

Table 2-44 Application Error Codes

Error Details	Error Description	Error Causes
App Error Codes: E171 HTTP Error Code: 401 Error Type: OAUTH_TOKEN_VALIDATION_FAILURE	This error code indicates that validation of Oauth token failed at Ingress Gateway.	Oauth token passed with the request and is enclosed with the values which are incorrect to the configuration present at the Ingress Gateway.
App Error Codes: E172 HTTP Error Code: 408 Error Type: OAUTH_CERT_EXPIRED	This error code indicates that Oauth certificate expired is received at Ingress Gateway.	Indicates that all the attributes are present and in the correct format, but the token has expired with respect to ingress gateway configuration.
App Error Codes: E173 HTTP Error Code: 407 Error Type: OAUTH_MISMATCH_IN_KID	This error code indicates that Oauth miss match in KID at Ingress Gateway.	Incorrect issuer and KID value are included in the request.
App Error Codes: E174 HTTP Error Code: 406 Error Type: OAUTH_PRODUCER_SCOPE_NOT_PRESENT	This error code indicates that producer scope not present in the Oauth token received at Ingress Gateway.	The producer scope is not present in the token of the request.
App Error Codes: E175 HTTP Error Code: 405 Error Type: OAUTH_PRODUCER_SCOPE_MISMATCH	This error code indicates that producer scope present in the Oauth token do not match with the producer scope configured in Ingress Gateway.	The producer scope sent through the token either has a wrong format, is used incorrectly, or incorrect value is present.
App Error Codes: E177 HTTP Error Code: 404 Error Type: OAUTH_PRODUCER_PLMNID_MISMATCH	This error code indicates that the received Oauth token do not match with producer PLMN configured in Ingress Gateway.	The configured PLMN ID in ingress gateway is not matching with the PLMN ID received through the Oauth token.

Table 2-44 (Cont.) Application Error Codes

Error Details	Error Description	Error Causes
App Error Codes: E178 HTTP Error Code: 403 Error Type: OAUTH_AUDIENCE_NOT_PRE SENT_OR_INVALID	This error code indicates that the received Oauth token does not contain an audience or invalid audience at Ingress Gateway.	Token is passed with audience parameter containing only correct nfType but in a different case. An audience parameter with missing or incorrect values for NFType and NfInstanceId can also cause this error.
App Error Codes: E179 HTTP Error Code: 402 Error Type: OAUTH_TOKEN_INVALID	This error code indicates that invalid Oauth token received at Ingress Gateway.	Indicates that the Oauth token received at Ingress Gateway is not valid, or not in the correct format.
App Error Codes: E180 HTTP Error Code: 400 Error Type: OAUTH_TOKEN_ABSENCE	This error code indicates that request received at ingress gateway with no Oauth token or empty Oauth token.	Oauth token is missing in the request.

To Maintain

If there is an expiry of configured public key certificate or an addition of new certificate for different NRF then change the existing configurations like:

- Delete an existing secret and create a new secret with updated public key certificate. To delete a secret, run the following command:
`kubectl delete secret oauthsecret -n ocudr`
- Send the certificate configuration update request using [CNCC GUI](#). The request should include the keyIdList and instanceIdList with new certificates.

2.35 Multiple Site Redundancy

Overview

UDR or SLF supports multiple site redundancy using the cnDBTier multi-site deployment model. All sites of UDR are active and can take traffic. The replication of database is handled by configuring DBTier for multiple site deployment.

Each UDR site has its own copy of the configuration database and subscriber database. All databases are replicated across all sites. Each site accesses its own configuration database. Each site access same subscriber databases, which are replicated.

Any updates of subscriber database are immediately replicated across all sites and hence, the redundancy is achieved.

When local site is down, UDR application fails due to database not being available. UDR starts failing the readiness checks and after configurable number of times, UDR deregisters from NRF. UDR does not connect to remote site of database.

Once the local database connectivity is restored, UDR application starts processing the traffic and also registers again with NRF.

Other sites where database is up, the UDR application has no impact.

To Enable and Configure

To enable and configure multiple site redundancy, see the 'Create Database User or Group' section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

2.36 Migrating Subscriber Data from 4G UDR to 5G UDR

Overview

Converged UDR provides the capabilities of both 5G UDR and 4G OCUDR. It supports migrating subscriber data from 4G OCUDR to the cloud-native UDR (cnUDR or 5G UDR). The cnUDR provides functionality such as migration tools and On-Demand Migration features to migrate the 4G policy subscriber data from 4G OCUDR to cnUDR.

There are two ways to migrate subscribers data from 4G OCUDR to the cloud native UDR (cnUDR):

- Migration Tool : Provides the capability to migrate bulk of subscribers from 4G OCUDR to cnUDR. For more information, see [Migration tool](#).
- On-Demand Migration: You can migrate 4G OCUDR subscriber information to 5G UDR, provided the 4G UDR subscriber information is not already using Migration Tool or Subscriber Bulk Import Tool. On-Demand Migration triggers on provisioning or signaling request. For more information, see [On-Demand Migration](#).

Note

It is recommended to use Migration Tool in combination with On-Demand Migration feature to ensure that the signaling and provisioning traffic are migrated to cnUDR without data loss. To rollback the exported subscriber data to 4G OCUDR, see [Support for EXML Format](#) section.

To Observe

See [Migration Metrics](#) and [Alert Details](#) section for more information on metrics and alerts respectively.

For troubleshooting information, see "nudr-migration" section in *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide*.

2.36.1 Migration Tool

The migration tool is an external tool that is used to migrate the subscriber data from source UDR (4G OCUDR) to target UDR (cnUDR). Migration tool is delivered as a separate Helm chart and you must install it manually. The migration tool supports migration of the subscribers in ranges, based on the configuration provided in the Helm chart.

The migration tool uses the following sequence to migrate the subscriber from 4G OCUDR to cnUDR:

- Migration tool connects to 4G OCUDR using diameter gateway.
- Migration tool reads the subscriber information from 4G OCUDR and send the provisioning requests to cnUDR through Ingress Gateway provisioning.

- On completion of migration, creates a report with INFO log level. The report includes summary of the subscriber migration.

Prerequisites

The prerequisites to use the Migration tool are as follows:

- Before you install the yaml file, 4G OCUDR and cnUDR must be in active and running state to start the migration process.
- The XSI IP network of 4G MP nodes must be accessible from kubernetes nodes, where you are deploying.
- Enable subscriber auto create as true in the nudr-dr section of the values yaml file.

2.36.1.1 Installing the Migration Tool

Migration tool is delivered as a separate Helm chart and you must install it manually. The migration tool supports migration of the subscribers in ranges, based on the configuration provided in the Helm chart. To install the migration tool, you must edit the custom value.yaml file that has configuration details of the migration tool.

Configuration changes in the custom value.yaml file is as follow:

```
migration:
startRange: "5556555531"
endRange: "5556555541"
realm: "oracle.com"
identity: "migration.oracle.com"
nodePort: 3868
nodeHost: "ocudr-nudr-migration-service"
```

```
fourG:
realm: "oracle.com"
identity: "tekelec.oracle.com"
nodePort: 3868
nodeHost: "10.196.7.118"
maxConnectionRetry: 10
```

```
gateway:
realm: "oracle.com"
identity: "nudr.oracle.com"
port: 3868
```

```
keyType: "msisdn"
```

Table 2-45 Migration Tool Parameters

Parameters	Description
migration.startRange/endRange	It is the subscriber range value that is migrated from 4G UDR to 5G UDR.
migration.realm	It is the realm of the migration tool that is used by the diameter gateway to establish connection to the migration tool.

Table 2-45 (Cont.) Migration Tool Parameters

Parameters	Description
migration.identity	It is the identity of the migration tool that is used by the diameter gateway to establish connection to the migration tool.
migration.nodePort	It is the nodePort of the migration tool that is used by the diameter gateway to establish connection to the migration tool.
migration.nodeHost	It is the nodeHost of the migration tool that is used by the diameter gateway to establish connection to the migration tool.
fourG.realm	It is the realm of the 4G that is used by the diameter gateway to establish connection to the 4G. This must be configured as the 4G server details.
fourG.identity	It is the identity of the 4G that is used by the diameter gateway to establish connection to the 4G. This must be configured as the 4G server details.
fourG.nodePort	It is the nodePort of the 4G that is used by the diameter gateway to establish connection to the 4G. This must be configured as the 4G server details.
fourG.nodeHost	It is the nodeHost of the 4G that is used by the diameter gateway to establish connection to the 4G. This must be configured as the 4G server details.
fourG.maxConnectionRetry	Maximum number of retry for migration tool to connect to the diameter gateway.
gateway.realm	It is the realm of the diameter gateway.
gateway.identity	It is the identity of the diameter gateway.
gateway.port	It is the port of the diameter gateway.
keyType	It is key type that must be migrated from the configured range.

Migration Tool Support for Multiple Pods

Update the following configuration as required to support the migration tool for multiple pods.

```
deployment:
# Replica count for deployment
replicaCount: 1
```

Note

It is recommended to use ReplicaCount as 1 when migrating less than 10 subscribers. The replicaCount must be less than the difference between startRange and endRange.

In a multiple pod deployment the configured ranges are equally distributed among all the pods. When configuring multiple replicas make sure that the difference between startRange and endRange values are sufficient to distribute to all the pods.

Run the following command to install the migration tool:

```
helm install migration -n <namespace>
```

Run the following command to check whether the pod is running state without error:

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

```

NAME                                READY   STATUS    RESTARTS   AGE
fourgstub-fourg-stub-79468fbf8-xcwmv 1/1     Running   0           6d5h
migration-nudr-migration-tool-0       1/1     Running   0           7h20m
migration-nudr-migration-tool-1       1/1     Running   0           7h19m
migration-nudr-migration-tool-2       1/1     Running   0           7h19m

```

You can configure the migration tool using CNC Console. For more information, see [Migration Tool Configurations](#) and [Migration Tool Status](#).

2.36.1.2 Configuring Migration Tool

The following table list the parameters for initContainer and mainContainer used for installing Migration Tool.

Table 2-46 Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
global.dockerRegistry	Docker registry from where the images are pulled.	ocudr-registry.us.abc.com:5000/ocudr	NA
global.mysql.dbServiceName	DB service to connect	mysql-connectivity-service.occne-infra	NA
global.mysql.port	Port for DB Service Connection	3306	NA
global.serviceAccountName	The entire pod must be associated with the serviceaccount used for ocudr deployment.	ocudr-serviceaccount	NA
preInstall.image.name	Image name for the nudr_pre_migration_hook pod which will take care of DB and table creation for UDR deployment.	nudr_pre_migration_hook	NA
preInstall.root.logLevel	Log level for preinstall hook pod root level	WARN	Possible Values: WARN INFO DEBUG
preInstall.ocudr.logLevel	Log level for preinstall hook pod app log level	WARN	Possible Values: WARN INFO DEBUG

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
preInstall.image.tag	Image version for nudr_pre_migration_hook pod image	25.1.201	NA
postUpgrade.config.logLevel	Log level for postUpgrade hook pod	WARN	Possible Values: WARN INFO DEBUG
global.hookJobResources.limits.cpu	CPU limit for pods created k8s hooks/jobs created as part of UDR installation. This is applicable for Helm test job too.	1	NA
global.hookJobResources.limits.memory	Memory limit for pods created k8s hooks/jobs created as part of UDR installation. This is applicable for Helm test job too	1Gi	NA
global.hookJobResources.limits.ephemeral-storage	Ephemeral Storage limits for hook pods	1024Mi	Unit MB
global.hookJobResources.requests.cpu	CPU requests for pods created k8s hooks/jobs created as part of UDR installation. This is applicable for Helm test job too. The CPU to be allocated for hooks during deployment.	0.5	NA
global.hookJobResources.requests.memory	Memory requests for pods created k8s hooks/jobs created as part of UDR installation. This is applicable for Helm test job too. The memory to be allocated for hooks during deployment.	0.5Gi	NA
global.hookJobResources.requests.ephemeral-storage	Ephemeral Storage request for hook pods.	72Mi	Unit MB
global.hooksCommonConfig.tolerations	When global.tolerationSetting is ENABLED configure tolerations here. Example: - tolerations: - key: "key1" "Equal" value: "value1" - operator: effect: "NoSchedule"	[]	NA

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
global.hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: - v1 - v2
global.hooksCommonConfig.nodeSelector.nodeKey	NodeSelector key configuration for UDR hooks. This excludes the common service hooks When hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration will come into use.	' '	Valid key for a label for a particular node.
global.hooksCommonConfig.nodeSelector.nodeValue	NodeSelector Value configuration for UDR hooks. This excludes the common service hooks. When hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is in use.	' '	Valid value pair for the above key for a label for a particular node.
global.hooksCommonConfig.nodeSelector	NodeSelector configuration when hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion at micro service 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.
dbCredSecretName	DB Credential Secret Name	ocudr-secrets	NA
configServerEnable	Enable configuration updates using config REST APIs	true	true or false
customExtension.allResources.labels	Custom Labels that needs to be added to all the OCUDR k8s resources.	null	NA
customExtension.allResources.annotations	Custom Annotations that needs to be added to all the OCUDR k8s resources. Note: ASM related annotations to be added under ASM Specific Configuration section.	null	NA

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
customExtension.lbServices.labels	Custom Labels that needs to be added to OCUDR Services that are considered as Load Balancer type.	null	NA
customExtension.lbServices.annotations	Custom Annotations that needs to be added to OCUDR Services that are considered as Load Balancer type.	null	NA
customExtension.lbDeployments.labels	Custom Labels that needs to be added to OCUDR deployments that are associated to a Service which is of Load Balancer type.	null	NA
customExtension.lbDeployments.annotations	Custom Annotations that needs to be added to OCUDR deployments that are associated to a Service which is of Load Balancer type. Note: ASM related annotations to be added under ASM Specific Configuration section.	null	NA
customExtension.nonlbServices.labels	Custom Labels that needs to be added to OCUDR services that are considered as not load balancer type.	null	NA
customExtension.nonlbServices.annotations	Custom Annotations that needs to be added to OCUDR services that are considered as not load balancer type.	null	NA
customExtension.nonlbDeployments.labels	Custom Labels that needs to be added to OCUDR Deployments that are associated to a Service which is not of load balancer type.	null	NA

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
customExtension.nonlb Deployments.annotations	Custom Annotations that needs to be added to OCUDR deployments that are associated to a Service which is not of load balancer type. Note: ASM related annotations to be added under ASM Specific Configuration section.	null	NA
extraContainers	Configuration to enable debug tool container	DISABLED	DISABLED or ENABLED

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
extraContainerTpl	The template for extraContainer will be defined in this configuration. The debugContainer template defined here. This is applicable for all microservices under ocudr umbrella.	<pre> - command: - /bin/ sleep - infinity image: {{ .Values.global.dockerRegistry }}/ debug- tools:25.1.203 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" cpu: "0.5" memory: "4Gi" securityContext: allowPrivilegeEscalation: true capabilities: drop: - ALL </pre>	NA

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
		<pre> add: - NET_RAW - NET_ADMIN runAsUser: 1002 volumeMounts: - mountPath: /tmp/ tools name: debug-tools-dir </pre>	
extraContainersVolumesTpl	Debug container volume template	<pre> - name: debug- tools-dir emptyDir: medium: Memory sizeLimit: "4Gi" </pre>	NA
image.name	Docker Image name	nudr_migration	NA
image.tag	Tag of Image	25.1.201	NA
image.pullPolicy	This setting will tell if image need to be pulled or not	IfNotPresent	Possible Values: Always IfNotPresent Never
configservice.port.http	Config service port	5001	Valid Port
migration.startRange	Defines the range of subscriber data in which migration should start	"5556555531"	NA
migration.endRange	Defines the range of subscriber data in which migration should end	"5556555541"	NA
migration.realm	It is the migration tool realm	"oracle.com"	NA
migration.identity	It is the migration tool identity	"migration.oracle.com"	NA
migration.nodePort	It is the migration tool port	3868	Valid Port
migration.nodeHost	It is the migration tool host	"ocudr-nudr-migration-service"	Valid Port
migration.migrationPauseEnabled	Migration Pause Feature Enabled	false	true/false
fourG.realm	Fourg realm details	"oracle.com"	NA

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
fourG.identity	Fourg identity details	"tekelec.oracle.com"	NA
fourG.nodePort	Fourg nodeport details	3868	Valid Port
fourG.nodeHost	Fourg nodehost details	"10.196.7.118"	Valid IP
fourG.maxConnectionRetry	Fourg maxConnection Retry Times	10	NA
gateway.realm	It is the realm of Diameter Gateway	"oracle.com"	NA
gateway.identity	It is the port of Diameter Gateway	"3868"	"3868"
gateway.port	It is the identity of Diameter Gateway	"nudr.oracle.com"	NA
keyType	Defines the range of subscriber data in which migration should start.	msisdn	NA
batchConfig.chunkSize	This is engineering configurations.	2500	NA
batchConfig.throttlelimit	This is engineering configurations.	100	NA
batchConfig.coreThreadPool	This is engineering configurations.	10	NA
batchConfig.maxThreadPool	This is engineering configurations.	10	NA
batchConfig.queueCapacity	This is engineering configurations.	30000	NA
batchConfig.writerCoreThreadPool	This is engineering configurations.	10	NA
batchConfig.writerMaxThreadPool	This is engineering configurations.	10	NA
batchConfig.writerQueueCapacity	This is engineering configurations.	30000	NA
service.type	It is the UDR service type. The kubernetes service type for exposing UDR deployment. Note: Suggested to be set as ClusterIP (default value) always.	ClusterIP	Possible Values- ClusterIP NodePort LoadBalancer
service.port.http	The HTTP port to be used in notify service to receive signals from nudr-migration-service pod.	5001	NA
service.port.management	Bulk import Management port. The actuator management port to be used for notify service.	9000	NA

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
service.customExtension.labels	Custom Labels that needs to be added to nudr-migration specific Service. This can be used to add custom label(s) to nudr-diameterproxy Service.	null	NA
service.customExtension.annotations	Custom Annotations that needs to be added to nudr-migration specific Services. This can be used to add custom annotation(s) to nudr-migration Service.	null	NA
deployment.replicaCount	Number of replicas for export tool	1	NA
deployment.customExtension.labels	Custom Labels that needs to be added to nudr-migration specific Deployment. This can be used to add custom label(s) to nudr-migration Deployment.	null	NA
deployment.customExtension.annotations	Custom Annotations that needs to be added to nudr-migration specific Deployment. This can be used to add custom annotation(s) to nudr-migration Deployment.	null	NA
hikari.poolsize	It is the Mysql connection pool size. The hikari pool connection size to be created at start up.	10	NA
hikari.connectionTimeout	It is the MYSQL connection timeout.	1000	Unit: Milliseconds
hikari.minimumIdleConnections	It is the minimum idle MYSQL connections maintained.	2	Valid No, less than poolsize configured
hikari.idleTimeout	It is the Idle MYSQL connections timeout.	1000	Unit: Milliseconds
hikari.queryTimeout	It is the Query Timeout for a single DB query. If set to -1, there will be no timeout set for DB queries.	-1	Unit: Seconds
logging.level.root	Log level of the export tool pod.	WARN	Possible Values - WARN INFO DEBUG

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
logging.level.ocudr	It is the application Log Level. App Log level of the migration tool pod.	WARN	Possible Values - WARN INFO DEBUG
resources.requests.cpu	CPU Allotment for nudr-notify-service pod. The CPU to be allocated for migration service pod during deployment.	2	NA
resources.requests.memory	Memory allotment for nudr-bulk-import pod. The memory to be allocated for nudr-migration pod during deployment.	2Gi	NA
resources.requests.logStorage	Log storage for ephemeral storage allocation request.	50	Unit: MB
resources.requests.criticalStorage	Critical storage for ephemeral storage allocation request.	2	Unit: MB
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
resources.limits.criticalStorage	Critical storage for ephemeral storage allocation limit.	2	Unit: MB
readinessProbe.initialDelaySeconds	Configurable wait time before performing the first readiness probe by the kubelet. Note: Do not change this value. If there is delays in pod coming up and probe is killing the pod then you must consider tuning these parameters.	30	Unit: Seconds
readinessProbe.periodSeconds	Time interval for every readiness probe check. Note: Do not change this value. If there is delays in pod coming up and probe is killing the pod then you must consider tuning these parameters.	3	Unit: Seconds

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
livenessProbe.initialDelaySeconds	Configurable wait time before performing the first liveness probe by the kubelet. Note: Do not change this value. If there is delays in pod coming up and probe is killing the pod then you must consider tuning these parameters.	30	Unit: Seconds
livenessProbe.periodSeconds	Time interval for every liveness probe check. Note: Do not change this value. If there is delays in pod coming up and probe is killing the pod then you must consider tuning these parameters.	3	Unit: Seconds
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
nodeSelector.nodeKey	Micro service level NodeSelector key configuration When helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. This configuration do not depend on nodeSelection flag. once configured this will be used for all microservices.	' '	Valid key for a label for a particular node

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
nodeSelector.nodeValue	Global level NodeSelector Value configuration When helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration is used. This configuration do not depend on nodeSelection flag. once configured this will be used for all microservices.	' '	Valid value pair for the above key for a label for a particular node
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
maxConcurrentPushed Streams	Maximum number of concurrent requests that is pushed per destination	6000	NA
maxRequestsQueuedPerDestination	Maximum number of requests that is queued per destination	5000	NA
maxConnectionsPerDestination	Maximum connection allowed per destination	10	NA
maxConnectionsPerIp	Maximum connection allowed per IP	10	NA
requestTimeout	Client Request timeout	5000	Unit: Milliseconds
connectionTimeout	Client connection timeout	10000	NA
idleTimeout	Client connection idle timeout	0	NA
dnsRefreshDelay	Server DNS refresh delay	120000	NA
pingDelay	Server ping delay	60	NA
connectionFailureThreshold	Client connection failure threshold	10	NA
jettyServiceMeshCheck	For ASM deployment jetty service mesh check	false	true or false
diameterServer	Diameter Server flag	true	true or false
diameterPort	Diameter port	"3868"	Valid Port
deleteSourceUDR	Delete content from source UDR	false	true or false
sourceEndPointURL	4G endpoint URL	"http://<4GRESTURL>:PORT"	NA
udrServiceBaseURL	5G UDR Service endpoint	"http://ocudr-ingressgateway-prov:80"	NA

Table 2-46 (Cont.) Migration Tool Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)
gracefulShutdown.gracePeriod	Graceful Shutdown time limit	30s	Unit: Seconds
internalTracingEnabled	The flag enables jaeger tracing for the service	false	true or false

2.36.1.3 Using the Migration Tool

You can use the migration tool only if:

- There exists an active source such as 4G connections are enabled and diameter gateway is running and destination source such as 5G UDR is up and running.
- The diameter gateway connections on 4G UDR are configured. It means the peer node information and connections are created with diameter gateway.
- Subscribing Client Permission is available for migration identity on 4G UDR for connecting to the migration tool.
- It is installed as per the installation process shared at [Installing the Migration Tool](#).

You can verify 5G UDR databases for all the 4G UDR records once the migration job completes. You can observe its metrics that are pegged in the Grafana dashboard.

Below sample report shows the success and failure records count in the INFO logs of nudr-migration job.

Migration Tool Summary Report:

```
| Time taken for processing: 0min 11sec |
| Per Pod Number of records: 10 |
| Per Pod Number of source UDR success: 10 |
| Per Pod Number of source UDR subscriber not found: 0 |
| Per Pod Number of target UDR keys already exists: 0 |
| Per Pod Number of source UDR failure: 0 |
| Per Pod Number of target UDR success: 10 |
| Per Pod Number of target UDR failure: 0 |
| List of failed source UDR keys Per Pod: |
| List of failed subscriber not exist keys at source Per Pod: |
| List of keys already exists in target UDR Per Pod: |
| List of failed target UDR keys Per Pod: |
```

The success and failure records counts are published in the INFO logs of nudr-migration job.

In the following cases, the UDR key list logs frequently, if the failed UDR key list size is equal to 5000 UDR keys:

- If the data migration from 4G UDR to 5G UDR fails due to target 5G UDR INFO log. For example: List of failed target UDR keys: 63612000056, 63612000057.
- If the data migration from 4G UDR to 5G UDR fails due to source 4G UDR INFO log. For example: List of failed source UDR keys: 63612000058 63612000059.

- If the data migration from 4G UDR to 5G UDR fails due to subscriber does not exist at source 4G UDR INFO log. For example: List of failed subscriber keys does not exist at source: 63612000060 63612000061.
- If the data migration from 4G UDR to 5G UDR fails due to already existing keys in target 5G UDR INFO log. For example: List of keys already existing in target UDR: 63612000062 63612000063.

Migration Tool Upgrade and Rollback

Upgrade and rollback of migration tool is not supported. Migration tool must be uninstalled when the migration job is completed.

Run the following command to uninstall the migration tool:

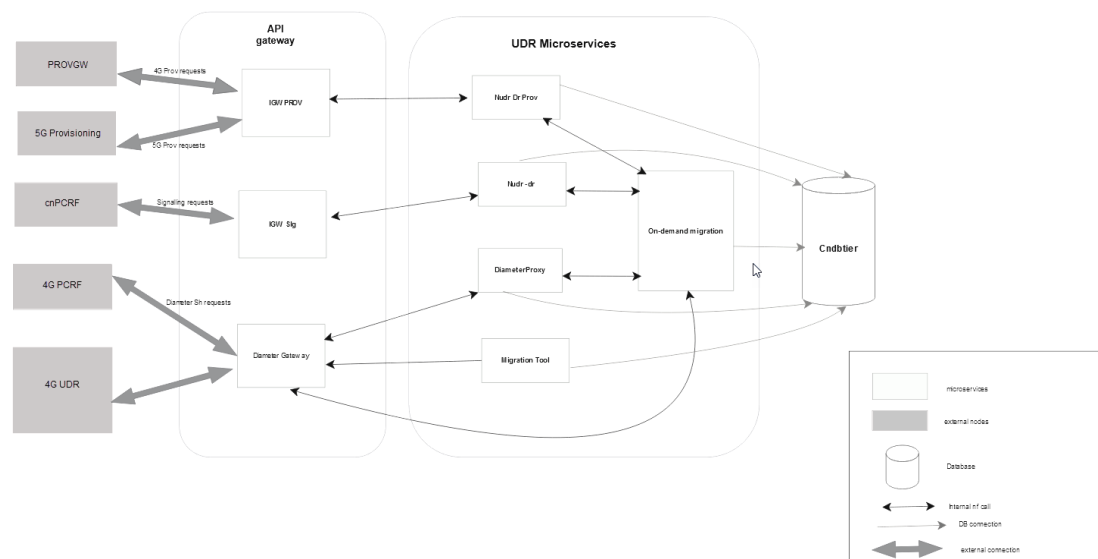
```
helm uninstall <release name> -n <namespace>
```

2.36.2 On-Demand Migration

The on-demand migration feature is supported as part of Converged Policy database solution of UDR. By enabling this feature, you can migrate 4G UDR subscriber information to 5G UDR, provided the 4G UDR subscriber information is not already provisioned on 5G platform. For more information on converged quota, see [Converged Quota Support for UDR](#).

This following diagram depicts the call flow between microservices when on-demand migration is triggered from the external clients such as cnPCRF, PCF and so on. When the on-demand migration requests reaches the microservices, it uses the Diameter Gateway to connect to 4G UDR and migrates the data to cnUDR. The diagram shows on-demand migration is triggered from various microservices when different types of requests are processed.

Figure 2-29 On-Demand Migration



On-demand migration triggers when any one of the following events occur:

- **Provisioning Event:** Provisioning Gateway sends the provisioning requests such as read or update subscriber information to cnUDR. If the on-demand migration feature is enabled

and if the subscriber does not exist on cnUDR, the subscriber information is read from 4G UDR and provisioned on cnUDR before sending back the response to the Provisioning Gateway. cnUDR reads and provisions the 4G policy quota data using the data model defined in [Table 2-20](#) and supports:

- Provisioning request for read and update using SOAP/XML provisioning interface. For more information, see *Oracle Communications Cloud Native Core, Provisioning Gateway Interface Specification Guide*.
- GET, PUT, and PATCH operations using REST/JSON provisioning interface. For more information, see in *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.
- **Nudr Event:** Based on a 5G session request initiated by PCF and an Nudr request received on cnUDR, the subscriber information is read from 4G UDR and provisioned on cnUDR.
 - Provided the on-demand migration feature is enabled on cnUDR
 - Subscriber information does not exist on cnUDR

The response to PCF or Converged Policy is sent after the migration of the subscriber information.

- **Sh-event:** Based on request (Diameter-Sh) received from PCRF to access the subscriber profile or entities using the UDR/PUR/SNR requests, the subscriber information is read from 4G UDR and provisioned on cnUDR
 - Provided the on-demand migration feature is enabled on cnUDR
 - Subscriber information does not exist on cnUDR

The response to PCRF is sent after the successful migration of the subscriber.

Note

On-demand migration can be enabled or disabled for each Provisioning Event, Nudr Event, and Sh-event independently.

Deleting Source UDR Subscriber Information

You can delete the source 4G UDR subscriber information after migrating the subscriber information from source 4G UDR to 5G UDR. You can disable or enable this feature by setting the `deleteSubscriberOnSource` parameter to true or false. The default configuration is false.

Note

You may experience delay in response from cnUDR if the on-demand migration feature is enabled.

To Enable

Pre-requisite: The on-demand migration service reads subscriber information from 4G UDR using Diameter connection(Sh) with 4G UDR. To read subscriber information from 4G UDR, it is important to configure 4G UDR with peer Diameter connections of cnUDR.

To enable on-demand migration service:

1. In the global values.yaml file, under nudr-dr service section, set the `ondemandmigration` property as 'true' and run the Helm upgrade command. By default, it is set as false.
2. Upon setting the `ondemandmigration` property as 'true', data is fetched from 4G UDR and provisioned on cnUDR when any of the event occurs (provisioning event, Nudr event or sh-event).

Note

After enabling the `ondemandmigration` parameter, the on-demand migration service runs as a separate service. The `nudr-dr` and `diameter-proxy` microservice uses it for handling subscriber migration.

3. After migrating the subscriber information to cnUDR, set the `deleteSubscriberOnSource` parameter to 'true' to delete that subscriber from 4G UDR.

Figure 2-30 Enabling `ondemandmigration` and `deleteSubscriberOnSource` parameter

```
ondemandMigration:
  enabled: &onDemandMigrationEnabled false
  diamRealm: "oracle.com"
  diamIdentity: "ondemand-migration.nudr.oracle.com"
  deleteSubscriberOnSource: false
  restEndpoint: http://source-udr-rest-endpoint
sourceUdr:
  realm: "tekelec1.com"
  identity: "local1.tekelec1.com"
  hostIp: ""
  hostPort: "3868"
```

Note

You can set the above parameter either from the CNC Console or the yaml file. If you opt to use yaml file then ensure that the `nudr_config` parameter is not enabled. After updating the yaml file, run the Helm upgrade command.

Range Support for On-Demand Migration

Range support for On-demand migration helps to migrate the 4G subscriber data to 5G UDR in a phase manner. You can initiate migration for small range of subscriber data and validate the call flows. Once the migration is successful, you can migrate another small range of subscriber data in a phase manner.

The below fields are added to introduce range support for on-demand migration:

- **Key Type:** This defines the type of key for the key range. The key type can either be Mobile Station Integrated Services Digital Network (MSISDN) or International Mobile Subscriber Identity (IMSI). Key type is not validated, if the default key range is set.

- **Key Range:** This defines the range of subscribers for on-demand migration. Multiple key ranges are not allowed. The default value is 000000-000000.

By default, if there are no configuration changes, on-demand migration works for all key type and key range.

To Configure

Perform the below steps to configure the on-demand migration feature:

- Edit the Helm config file while installing UDR. This flag is not available on CNC Console.
- Configure Diameter-Sh peers on the Helm charts or CNC Console. 5G UDR uses these Diameter-Sh connections with 4G UDR to read its subscribers information.

Using this configuration, you can also delete the subscriber information migrated to 5G UDR from source 4G UDR.

Configuring Range Support

To configure the range support, you must set the key type and key range in the global section of the custom-values.yaml file. on-demand migration triggers only if the range is set and the key is in the range value.

```
global:
# Key type and range for ondemand qouta migration
key:
type: msisdn
range: 000000-000000
```

Configuring Range Support using REST API

You can perform the PUT operation on the `http://nudr-config-service:PORT/ nudr-config/v1/udr.global.cfg/GLOBAL` to configure the range support. For more information, see Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide.

```
{
  "dbServiceName": "mysql-connectivity-service.occne-ndb",
  "udrServices": "All",
  "udsfEnabled": false,
  "ingressHttpPort": "",
  "ingressHttpsPort": "",
  "snssai": "2-FFFFFF",
  "dnn": "dnn1",
  "autoCreate": true,
  "autoEnrolOnSignalling": true,
  "etagEnabled": true,
  "sbiCorrelationInfoEnable": false,
  "consumerNF": "PCF,UDM,NEF",
  "nfInstanceId": "5a7bd676-ceed-44bb-95e0-f6a55a328b03",
  "vsaDefaultBillingDay": 1,
  "subscriberActivityEnabled": false,
  "addDefaultBillingDay": true,
  "enableControlledShutdown": true,
  "keyType": "imsi",
  "keyRange": "2013020001-2013020002",
  "nfType": "UDR",
```

```

    "nfFqdn": "udr001.oracle.com",
    "suppressNotificationEnabled": true,
    "configClientConnectTimeout": 1000,
    "configClientReadTimeout": 1000,
    "subscriberIdentifiers": {
      "msisdn": [],
      "imsi": [],
      "nai": [],
      "extid": []
    }
  }
}

```

Upgrade and Rollback Impacts for Range Support

When you upgrade UDR from previous version with default key type and key range, below scenarios can occur:

- If On-demand migration is enabled, then range support works for all key type and key range.
- If you modify the key type and key range through REST API, then the On-demand migration triggers only if the key type is valid and key is in the range value.

Note

Range support for on-demand migration is disabled when rollback is performed to a previous version.

2.36.2.1 Configuring On-Demand Migration

The following table list configurable parameters for **nudr-ondemand-migration microservice**.

Table 2-47 nudr-ondemand-migration microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
enabled	Flag to enable nudr-ondemand-migration microservice	false	NA	NA
image	Image from the Docker registry along with the image name and tag	ocudr-registry.us.abc.com:5000/ocudr/nudr_pre_migration_hook:25.1.201	NA	NA
image.name	Name of Docker Image	nudr_ondemand_migration	NA	NA
image.tag	Tag or version of Image	25.1.201	NA	NA
image.pullPolicy	This setting indicates if need to be pulled or not	IfNotPresent	Possible Values: Always IfNotPresent Never	NA

Table 2-47 (Cont.) nudr-ondemand-migration 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
service.port.http	HTTP port used in nudr-ondemand-migration service	5001	NA	NA
service.port.https	HTTPS port used for nudr-ondemand-migration service	5002	NA	NA
service.port.management	Management port number used for nudr-ondemand-migration service	9000	NA	NA
service.customExtension.labels	Use to add custom label(s) to nudr-ondemand-migration service	null	NA	NA
service.customExtension.annotations	This can be used to add custom annotation(s) to nudr-ondemand-migration service	null	NA	NA
drservice.port.http	dr-service HTTP port required in nudr-ondemand-migration application	5001	NA	NA
drservice.port.https	dr-service HTTPS port required in nudr-ondemand-migration application	5002	NA	NA
deployment.replicaCount	Number of nudr-ondemand-migration pods maintained by replica set created with deployment	2	NA	NA
deployment.customExtension.labels	Used to add custom label(s) to nudr-ondemand-migration deployment	null	NA	NA
deployment.customExtension.annotations	To add custom annotation(s) to nudr-ondemand-migration deployment	null	NA	NA

Table 2-47 (Cont.) nudr-ondemand-migration microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
logging.level.root	Log level of the nudr-ondemand-migration server pod	WARN	Possible Values: WARN INFO DEBUG	NA
resources.requests.cpu	Number of CPUs allocated for nudr-ondemand-migration pod during deployment	3	NA	NA
resources.requests.memory	Memory allocated for nudr-ondemand-migration pod during deployment	4Gi	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-ondemand-migration pod	3	NA	NA
resources.limits.memory	Maximum memory allocated for nudr-ondemand-migration pod	4Gi	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 2-47 (Cont.) nudr-ondemand-migration 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
minReplicas	Minimum number of pods allocated for nudr-diameterproxy	2	NA	NA
maxReplicas	Maximum number of pods allocated for nudr-diameterproxy	4	NA	NA
serviceToCheck	Essential services on which INIT container performs readiness checks	nudr-drservice	Comma separated microservice names	NA
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
hikari.queryTimeout	Query timeout for a single database query. If you set the value to -1 there will be no timeout for database queries	-1	Unit: Seconds	NA
maxConcurrentPushedStreams	Maximum number of concurrent requests that is pushed per destination	6000	NA	NA
maxRequestsQueuedPerDestination	Maximum number of requests that is 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	NA	NA
idleTimeout	Client connection idle timeout	0	NA	NA
dnsRefreshDelay	Server DNS refresh delay	120000	NA	NA
pingDelay	Server ping delay	60	NA	NA

Table 2-47 (Cont.) nudr-ondemand-migration microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
connectionFailureThreshold	Client connection failure threshold	10	NA	NA
jettyserviceMeshCheck	For ASM deployment jetty service mesh check	false	true/false	NA
diameter.identity	Identity (FQDN) of nudr-ondemand migration in diameter messages	ondemand-migration.udr.oracle.com	String value	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 whether service level extra container need to be used or not	USE_GLOBAL_VALUE	Allowed Values: - ENABLED - DISABLED - USE_GLOBAL_VALUE	NA
serviceMeshCheck	Flag to check service mesh. 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
istioSidecarReadyUrl	Readiness URL configurable for side car Change 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

Table 2-47 (Cont.) nudr-ondemand-migration microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
nodeNameAffinity	Slave nodename in which ondemand pod should to run	NA	NA	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 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 then tolerations configuration in the same section are 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, then configure tolerations.	[]	Example: <pre>tolerations: - key: "key1" operator: "Equal" value: "value1" effect: "NoSchedule"</pre>	NA

Table 2-47 (Cont.) nudr-ondemand-migration microservice parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
helmBasedConfiguration NodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: - v1 - v2	NA
nodeSelector.nodeKey	NodeSelector key configuration at the microservice level when helmBasedConfiguration NodeSelectorApiVersion 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 helmBasedConfiguration NodeSelectorApiVersion 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 helmBasedConfiguration NodeSelectorApiVersion 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
maxUnavailable	Number of pods that can be terminated during disruptions	1	NA	NA
gracefulShutdown.grace Period	It is the graceful shutdown time limit	30s	Unit: Seconds	NA

2.37 Subscriber Bulk Import Tool

This section describes the Subscriber Bulk Import Tool feature.

2.37.1 Subscriber Bulk Import Tool

The Subscriber Bulk Import tool has three containers: a main container, xmlToCsv container, and pdbiToCsv container.

To support EIR functionality, pdbiToCsv container is added to the bulk import tool.

The main container is used to provision 5G UDR subscribers data in bulk. It is also used to:

- Read the subscriber data from a CSV file in a pre-determined format and then process the records.

Note

The CSV file is located in a specified directory on the pod.

- Process each record and send requests to UDR for provisioning the subscribers
- Support PCF subscriber data (5G PCF and VSA), SLF, and EIR subscriber data
- Support Subscribers CREATE, MODIFY, and DELETE operations

The xmlToCsv container is used to convert subscriber records in iXML format to CSV format. The iXML format is same as defined for 4G UDR. This container:

- Reads the ixml subscriber data from the ixml file in a specified directory path on the pod.
- Converts ixml subscriber information to JSON format and generates CSV file as output.
- Uses the converted CSV file to provision subscribers on 5G UDR in PCF-VSA provisioning requests
- Supports different types of operations: CREATE, MODIFY, DELETE, and PATCH.

The pdbiToCsv container is used to convert PDBi format to CSV format. This container:

- Reads the PDBi record from the pdbi file in a specified directory path on the pod.
- Converts PDBi record to JSON format and generates CSV file as output.
- Uses the converted CSV file to provision subscribers on 5G EIR.
- Supports different types of operations: MODIFY, DELETE, and DELETE-KEY.

Note

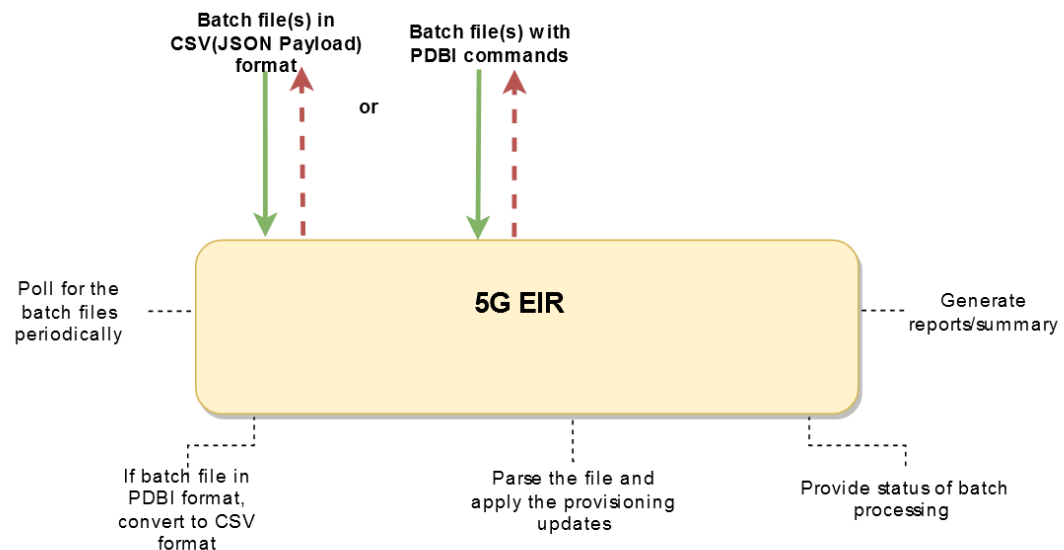
- The Bulk Import tool uses PVC (Persistent Volume Claim) and PV (Persistent Volume) that is created before deploying the tool. For more information about creating PVC and PV, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- You are required to manually copy all the CSV files to the PV mountpath inside the container, which is **/home/udruser/import/**.

Support for 5G EIR Bulk Provisioning

5G EIR supports bulk provisioning of subscriber data by using the batch files in both CSV and PDBi format. This feature provides the capability of processing the batch files containing EAGLE Provisioning Application Processor (EPAP) supported PDBi provisioning commands

that enables deployment of 5G EIR in stand alone mode and also in conjunction with the Oracle offered legacy EAGLE EPAP based 3G and 4G EIR deployments.

Figure 2-31 5G EIR Bulk Provisioning



Bulk Import File Format for PDBI Commands

PDBI provides commands that communicate provisioning information from the customer database to the Provisioning Database (PDB) in the Active Provisioning Database Application (PDBA) in an EAGLE. The customer can execute provisioning commands using a provisioning application. The provisioning application uses the PDBI request and response messages to communicate with the EPAP PDBA over the customer network.

PDBI command syntax:

```
sample_msg(field1 #, field2 #, [field3 <yes/no>], field4 <0..255>)
```

Example:

```
ent_eir(iid 1, imei 355171053475189, white yes, gray yes, imsi
111111111111115, imsi 111111111111116)
ent_eir(iid 2, imei 11111110000001, imsi 111111111111111, black yes)
ent_eir(iid 3, imei 11111110000002, msisdn 1111111114, black yes)
dlt_eir(iid 4, imei 355171053475189)
```

PDBI commands converted to CSV format:

```
keyType,keyValue,operationType,nfType,jsonPayload
IMEI,11111110000001,MODIFY,EIR,"{"cmd":"ent_eir(iid 2, imei 11111110000001,
imsi 111111111111111, black yes)","profile-data":{"imei":
["11111110000001"],"imsi":["111111111111111"],"equipment-status":
{"black":true}}}"
IMEI,11111110000002,MODIFY,EIR,"{"cmd":"ent_eir(iid 3, imei 11111110000002,
```

```
msisdn 1111111114, black yes)","profile-data":{"imei":
["111111100000002"],"msisdn":["1111111114"]},"equipment-status":
{"black":true}}"
IMEI,355171053475189,MODIFY,EIR,"{"cmd":"ent_eir(iid 1, imei 355171053475189,
white yes, gray yes, imsi 111111111111115, imsi 111111111111116)","profile-
data":{"imei":["355171053475189"],"imsi":
["111111111111115","111111111111116"]},"equipment-status":
{"white":true,"grey":true}}"
IMEI,355171053475189,DELETE,EIR,"{"cmd":"dlt_eir(iid 4, imei
355171053475189)}"
```

Example for IMEISV:

```
ent_eir(iid 3, imei 355171053475189, svn 01, white yes, gray yes, imsi
111111111111115, imsi 111111111111116)
dlt_eir(iid 6, imei 355171053475189, svn 01)
```

Different PDBI commands are given below:

- PDBI commands supported:
 - ent_eir
 - upd_eir
 - dlt_eir
- PDBI commands not supported:
 - ent_sub
 - upd_sub
 - dlt_sub
 - ent_entity
 - upd_entity
 - dlt_entity

To Observe

See [Bulk Import Metrics](#) for more information on metrics respectively.

2.37.1.1 Subscriber Bulk Import Operations

The Subscriber Bulk Import tool supports following operation types:

- **CREATE:** Allows to create subscribers with all keys mentioned in the payload. The PCF create and SLF create request payloads are available in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.
- **UPDATE:** Allows to update the existing subscribers with new data mentioned in the **subs_data** column.

Note

While using this operation, you should provide complete subscriber information as it replaces the old information with new data.

- **DELETE:** Allows to delete complete subscriber information with all keys attached to the subscriber.
- **PATCH:** Updates the existing subscribers with new data in the payload. It updates only particular field or element.

Note

For information about bulk import tool keyTypes, see [Sample Bulk Import File in CSV Format](#).

The Bulk Import tool supports following NF Types:

- **PCF:** Using this tool, you can provision all 5G PCF data in bulk for all resource types: AM data, SM Data, and UEPolicySet. For more information, see the **Operations supported for PCF Data** section in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.
- **SLF:** You can create SLF Subscribers with all keys and SLFGroupName in the payload. For more information, see the **Operations supported for SLF Data** section in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Note

The Bulk Import tool does not support SLFGroupName provisioning. You can provision this using REST APIs or CNC Console GUI.

- **PCF-VSA:** You can provision all 4G policy data in bulk for all entity types: subscriber, quota, state, and dynamicQuota. For more information, see the **Operations supported for Policy Data** section in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.
- **EIR:** You can create EIR Subscribers. For more information, see the **Provisioning Operations for EIR Data** section in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

Table 2-48 API for Provisioning

NF Types	CREATE	UPDATE	DELETE	PATCH
PCF	Yes	Yes	Yes	No
SLF	No	Yes	Yes	No
PCF-VSA	Yes	Yes	Yes	Yes
EIR	Yes	Yes	Yes	No

2.37.1.2 Bulk Import Operations in iXML Format

UDR and subscriber export tool requires a specific set of profile keys to have an array data type as their value, the bulk import XML to CSV conversion modifies the keys that consist of a single string value on the XML file to be converted to a JSONArray on the output CSV file. For this mechanism to work, you must have following casing for these keys:

- MSISDN
- IMSI

- NAI
- EXTID
- AccountId
- Entitlement

The below table provides operation details of bulk import tool in iXML format.

Table 2-49 iXML Format for Bulk Import

Opertion Detail	iXML Format
Transaction block with single request of Entity Dynamic Quota (createEntity request)	<pre> <transaction> <txRequest id="1"> <create createEntityIfNotExist="true"> <key> <MSISDN>447731751965</MSISDN> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath>/ definition</xpath> </data> <content> <![[CDATA[<DynamicQuota name="SMB_TOP"><InstanceId>753940540_ 2021-01-25T21:02:31_00001</ InstanceId><Type>pass</ Type><Priority>1</ Priority><expirationdatetime>2021-02- 24T23:59:59</ expirationdatetime><purchasedatetime> 2021-01-25T21:02:31</ purchasedatetime><InitialTotalVolume> 99143682048</InitialTotalVolume></ DynamicQuota>]]> </content> </entity> </create> </txRequest> </transaction> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Transaction block with uploadField and deleteField requests	<pre> <transaction> <txRequest id="1"> <updateField clearAll="false"> <key> <MSISDN>1000000017</ MSISDN> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> <xpath>/ subscriber</xpath> </data> <fields> <field name="IMSI">30237011111111</field> <field name="BillingDay">23</field> </fields> </entity> </updateField> </txRequest> <txRequest id="2"> <deleteField clearAll="false"> <key> <MSISDN>1000000017</ MSISDN> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> <xpath>/ subscriber</xpath> </data> <fields> <field name="IMSI">302370111111121</field> </fields> </entity> </deleteField> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
	</txRequest> </transaction>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Transaction block with multiple requests of updateField , deleteField , and deleteFieldSet of State entity	<pre> <transaction> <txRequest id="1"> <updateField clearAll="false"> <key> <MSISDN>15141234567</ MSISDN> </key> <entity> <data> <name>State</ name> </data> </entity> </updateField> </txRequest> <txRequest id="2"> <deleteField clearAll="false"> <key> <MSISDN>15141234567</ MSISDN> </key> <entity> <data> <name>State</ name> </data> </entity> </deleteField> </txRequest> <txRequest id="3"> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
	<pre><deleteFieldSet> <key> <MSISDN>15141234567</ MSISDN> </key> <entity> <data> <name>State</name> </interface>XMLIMPORT</interface> <xpath>/state/ property[name='approved']</xpath> </data> </entity> </deleteFieldSet> </txRequest> </transaction></pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Transaction block request with all entities (createSubscriber along with Create request for Quota, DynamicQuota, and State)	<pre> <transaction> <txRequest id="1"> <createSubscriber inTx="false"> <key> <IMSI>184569547984229</IMSI> <MSISDN>33123654862</MSISDN> <AccountId>10404723525</AccountId> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><subscriber><field name="AccountId">10404723525</ field><field name="MSISDN">33123654862</ field><field name="IMSI">184569547984229</ field><field name="BillingDay">1</ field><field name="Entitlement">DayPass</ field><field name="Entitlement">DayPassPlus</ field></subscriber>]]> </content> </entity> </createSubscriber> </txRequest> <txRequest id="1"> <create create="true" createEntityIfNotExist="false" inTx="false"> <key> <IMSI>184569547984229</IMSI> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath></xpath> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><definition><version>1</ </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
	<pre> version><dynamicquota name="AggregateLimit"><Type>Roll- Over</Type><InstanceId>15678</ InstanceId><Priority>4</ Priority><InitialTime>135</ InitialTime><InitialtotalVolume>2000< / InitialtotalVolume><InitialinputVolum e>1500</ InitialinputVolume><InitialoutputVolu me>500</ InitialoutputVolume><InitialserviceSp ecific>4</ InitialserviceSpecific><ActivationDat eTime>2015-04-22T00:00:00-05:00</ ActivationDateTime><ExpirationDateTim e>2015-06-22T00:00:00-05:00</ ExpirationDateTime><InterimReportingI nterval>100</ InterimReportingInterval><Duration>10 </Duration></dynamicquota></ definition>]]> </content> </entity> </create> </txRequest> <txRequest id="1"> <create create="true" createEntityIfNotExist="false" inTx="false"> <key> <IMSI>184569547984229</IMSI> </key> <entity> <data> <name>Quota</name> <interface>XMLIMPORT</interface> <xpath></xpath> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><usage><version>3</version><quota name="AggregateLimit"><cid>9223372036 854775807</cid><time>3422</ time><totalVolume>1000</ totalVolume><inputVolume>980</ inputVolume><outputVolume>20</ outputVolume><serviceSpecific>12</ </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
	<pre> serviceSpecific><nextResetTime>2011-0 4-22T00:00:00-05:00</ nextResetTime><QuotaState / ><RefInstanceId /><Type / ><grantedInputVolume>0</ grantedInputVolume><grantedOutputVolu me>0</ grantedOutputVolume><grantedServiceSp ecific /><grantedTime / ><grantedTotalVolume>0</ grantedTotalVolume></quota></ usage>]]> </content> </entity> </create> </txRequest> <txRequest id="1"> <create create="true" createEntityIfNotExist="false" inTx="false"> <key> <IMSI>184569547984229</IMSI> </key> <entity> <data> <name>State</name> <interface>XMLIMPORT</interface> <xpath></xpath> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><state><version>1</ version><property><name>mcc</ name><value>315</value></ property><property><name>expire</ name><value>2010-02-09T11:20:32</ value></ property><property><name>approved</ name><value>yes</value></property></ state>]]> </content> </entity> </create> </txRequest> </transaction> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Transaction block request with all Entities (createSubscriber along with updateFieldSet request for Quota, DynamicQuota, and State)	<pre> <transaction> <txRequest id="1"> <createSubscriber inTx="false"> <key> <IMSI>184569547984229</IMSI> <MSISDN>33123654862</ MSISDN> <AccountId>10404723525</AccountId> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><subscriber><field name="AccountId">10404723525</ field><field name="MSISDN">33123654862</ field><field name="IMSI">184569547984229</ field><field name="BillingDay">1</ field><field name="Entitlement">DayPass</ field><field name="Entitlement">DayPassPlus</ field></subscriber>]]> </content> </entity> </createSubscriber> </txRequest> <txRequest id="1"> <updateFieldSet create="true" createEntityIfNotExist="false" inTx="false"> <key> <IMSI>184569547984229</IMSI> </key> <entity> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
	<pre> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath></xpath> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><definition><version>1</ version><dynamicquota name="AggregateLimit"><Type>Roll- Over</Type><InstanceId>15678</ InstanceId><Priority>4</ Priority><InitialTime>135</ InitialTime><InitialtotalVolume>2000< / InitialtotalVolume><InitialinputVolum e>1500</ InitialinputVolume><InitialoutputVolu me>500</ InitialoutputVolume><InitialserviceSp ecific>4</ InitialserviceSpecific><ActivationDat eTime>2015-04-22T00:00:00-05:00</ ActivationDateTime><ExpirationDateTim e>2015-06-22T00:00:00-05:00</ ExpirationDateTime><InterimReportingI nterval>100</ InterimReportingInterval><Duration>10 </Duration></dynamicquota></ definition>]]> </content> </entity> </updateFieldSet> </txRequest> <txRequest id="1"> <updateFieldSet create="true" createEntityIfNotExist="false" inTx="false"> <key> <IMSI>184569547984229</IMSI> </key> <entity> <data> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
	<pre> <name>Quota</ name> <interface>XMLIMPORT</interface> <xpath></xpath> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><usage><version>3</version><quota name="AggregateLimit"><cid>9223372036 854775807</cid><time>3422</ time><totalVolume>1000</ totalVolume><inputVolume>980</ inputVolume><outputVolume>20</ outputVolume><serviceSpecific>12</ serviceSpecific><nextResetTime>2011-0 4-22T00:00:00-05:00</ nextResetTime><QuotaState / ><RefInstanceId /><Type / ><grantedInputVolume>0</ grantedInputVolume><grantedOutputVolu me>0</ grantedOutputVolume><grantedServiceSp ecific /><grantedTime / ><grantedTotalVolume>0</ grantedTotalVolume></quota></ usage>]]> </content> </entity> </updateFieldSet> </txRequest> <txRequest id="1"> <updateFieldSet create="true" createEntityIfNotExist="false" inTx="false"> <key> <IMSI>184569547984229</IMSI> </key> <entity> <data> <name>State</ name> <interface>XMLIMPORT</interface> <xpath></xpath> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
	<pre> </data> <content> <![CDATA[<?xml version="1.0" encoding="UTF-8"? ><state><version>1</ version><property><name>mcc</ name><value>315</value></ property><property><name>expire</ name><value>2010-02-09T11:20:32</ value></ property><property><name>approved</ name><value>yes</value></property></ state>]]> </content> </entity> </updateFieldSet> </txRequest> </transaction> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request to Create Subscriber	<pre> <createSubscriber> <key> <AccountId>10404723525</AccountId> <MSISDN>33123654862</MSISDN> <IMSI>184569547984229</IMSI> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> </data> <content> <![CDATA[<subscriber> <field name="AccountId">10404723525</field> <field name="MSISDN">33123654862</ field> <field name="IMSI">184569547984229</ field> <field name="BillingDay">1</field> <field name="Tier"></field> <field name="Entitlement">DayPass</ field> <field name="Entitlement">DayPassPlus</ field> </subscriber>]]> </content> </entity> </createSubscriber> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request to Update Subscriber	<pre> <updateSubscriber> <key> <MSISDN>33123654862</MSISDN> </key> <subscriber> <AccountId>10404723525</AccountId> <MSISDN>33123654862</MSISDN> <IMSI>184569547984229</IMSI> </subscriber> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> </data> <content> <![CDATA[<subscriber> <field name="AccountId">10404723525</ field> <field name="MSISDN">33123654862</ field> <field name="IMSI">184569547984229</ field> <field name="BillingDay">6</field> <field name="Tier">Silver</field> <field name="Entitlement">DayPass</ field> </subscriber>]]> </content> </entity> </updateSubscriber> </pre>
Non transaction request to Delete Subscriber	<pre> <deleteSubscriber> <key> <MSISDN>33123654862</MSISDN> </key> </deleteSubscriber> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with Create request to create Quota Entity	<pre> <create createEntityIfNotExist="false"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Quota</name> <interface>XMLIMPORT</interface> <xpath/> </data> <content> <![CDATA[<usage> <version>3</version> <quota name="AggregateLimit"> <cid>9223372036854775807</cid> <time>3422</time> <totalVolume>1000</totalVolume> <inputVolume>980</inputVolume> <outputVolume>20</outputVolume> <serviceSpecific>12</ serviceSpecific> <nextResetTime>2010-05-22T00:00:00-05:00</nextResetTime> </quota> </usage>]]> </content> </entity> </create> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with Create request to create State Entity	<pre> <create createEntityIfNotExist="false" > <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>State</name> <interface>XMLIMPORT</interface> <xpath/> </data> <content> <![CDATA[<state> <version>1</version> <property> <name>mcc</name> <value>315</value> </property> <property> <name>expire</name> <value>2014-02-09T11:20:32</value> </property> <property> <name>approved</name> <value>no</value> </property> </state>]]> </content> </entity> </create> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with Create request to create Dynamic Quota entity	<pre> <create createEntityIfNotExist="false" > <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath>/definition</xpath> </data> <content> <![CDATA[<DynamicQuota name="PDQ1"> <Type>Roll-over</Type> <InstanceId>15678</InstanceId> <Priority>4</Priority> <InitialTime>135</InitialTime> <InitialTotalVolume>2000</ InitialTotalVolume> <InitialInputVolume>1500</ InitialInputVolume> <InitialOutputVolume>500</ InitialOutputVolume> <InitialServiceSpecific>4</ InitialServiceSpecific> <activationdatetime>2015-03-09T11:20: 32</activationdatetime> <expirationdatetime>2015-04-9T11:20:3 2</expirationdatetime> <InterimReportingInterval>100</ InterimReportingInterval> <Duration>10</Duration> </DynamicQuota>]]> </content> </entity> </create> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with UpdateFieldSet request to update Quota entity	<pre> <updateFieldSet createEntityIfNotExist="false" create="false"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Quota</name> <interface>XMLIMPORT</interface> <xpath/> </data> <content> <![CDATA[<usage> <version>3</version> <quota name="AggregateLimit"> <cid>9223372036854775807</cid> <time>3422</time> <totalVolume>1000</totalVolume> <inputVolume>980</inputVolume> <outputVolume>20</outputVolume> <serviceSpecific>12</ serviceSpecific> <nextResetTime>2010-05-22T00:00:00-05:00</nextResetTime> </quota> </usage>]]> </content> </entity> </updateFieldSet> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with UpdateFieldSet request to update State entity	<pre> <updateFieldSet createEntityIfNotExist="false" create="false"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>State</name> <interface>XMLIMPORT</interface> <xpath/> </data> <content> <![CDATA[<state> <version>1</version> <property> <name>mcc</name> <value>315</value> </property> <property> <name>expire</name> <value>2014-02-09T11:20:32</value> </property> <property> <name>approved</name> <value>no</value> </property> </state>]]> </content> </entity> </updateFieldSet> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with UpdateFieldSet request to update DynamicQuota entity	<pre> <updateFieldSet createEntityIfNotExist="false" create="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath>/definition</xpath> </data> <content> <![CDATA[<DynamicQuota name="PDQ1"> <Type>Roll-over</Type> <InstanceId>15678</InstanceId> <Priority>4</Priority> <InitialTime>135</InitialTime> <InitialTotalVolume>2000</ InitialTotalVolume> <InitialInputVolume>1500</ InitialInputVolume> <InitialOutputVolume>500</ InitialOutputVolume> <InitialServiceSpecific>4</ InitialServiceSpecific> <activationdatetime>2015-03-09T11:20: 32</activationdatetime> <expirationdatetime>2015-04-9T11:20:3 2</expirationdatetime> <InterimReportingInterval>100</ InterimReportingInterval> <Duration>10</Duration> </DynamicQuota>]]> </content> </entity> </updateFieldSet> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with deleteFieldSet request to delete Quota entity	<pre> <deleteFieldSet> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Quota</name> <interface>XMLIMPORT</interface> <xpath/> </data> </entity> </deleteFieldSet> </pre>
Non transaction request with deleteFieldSet request for Quota to delete row Q2	<pre> <deleteFieldSet> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Quota</name> <interface>XMLIMPORT</interface> <xpath>/usage/quota[@name='Q2']</xpath> </data> </entity> </deleteFieldSet> </pre>
Non transaction request with deleteFieldSet request to delete State entity	<pre> <deleteFieldSet> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>State</name> <interface>XMLIMPORT</interface> <xpath/> </data> </entity> </deleteFieldSet> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with deleteFieldSet request for State entity to delete field mcc	<pre> <deleteFieldSet> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>State</name> <interface>XMLIMPORT</interface> <xpath>/state/property[name='mcc']</xpath> </data> </entity> </deleteFieldSet> </pre>
Non transaction request with deleteFieldSet request to delete Dynamic Quota Entity	<pre> <deleteFieldSet> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath/> </data> </entity> </deleteFieldSet> </pre>
Non transaction request with deleteFieldSet request for Dynamic Quota to delete row PDQ1	<pre> <deleteFieldSet> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath>/definition/ DynamicQuota[@name='PDQ1']</xpath> </data> </entity> </deleteFieldSet> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with updateField request to update a subscriber profile and set the value of the BillingDay field to 23 and the Tier field to Gold.	<pre> <updateField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> <xpath>/subscriber</xpath> </data> <fields> <field name="BillingDay">23</field> <field name="Tier">Gold</field> </fields> </entity> </updateField> </pre>
Non transaction request with updateField request to update a subscriber profile. The request performs the following functions: - Replaces all exist IMSI values with a single value - Deletes the specific MSISDN 14161112222 - Deletes all instances of the NAI - uupdates the BillingDay field - Deletes the Tier field if the existing value is set to Gold - Deletes the Custom5 field	<pre> <updateField clearAll="false"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> <xpath>/subscriber</xpath> </data> <fields> <field name="IMSI">302370111111111</field> <field name="MSISDN" delete="true">14161112222</field> <field name="NAI" delete="true"/> <field name="BillingDay">11</field> <field name="Tier" delete="true">Gold</field> <field name="Custom5" delete="true"/> </fields> </entity> </updateField> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with deleteField request to delete a subscriber profile and set the value of the BillingDay field to 23 and the Tier field to Gold	<pre> <deleteField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Subscriber</name> <interface>XMLIMPORT</interface> <xpath>/subscriber</xpath> </data> <fields> <field name="BillingDay">23</field> <field name="Tier">Gold</field> </fields> </entity> </deleteField> </pre>
Non transaction request with updateField request to update the inputVolume and the outputVolume fields in the Q1 Quota row in the Quota entity . The two rows are called Q1 exist, one row with a CID of 111 and another row with a CID of 222. The request is to update the instance with the CID of 111.	<pre> <updateField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Quota</name>Non Transaction Request with updateField request to <interface>XMLIMPORT</interface> <xpath>/usage/quota[@name='Q1' and cid='111']</xpath> </data> <fields> <field name="inputVolume">3000</ field> <field name="outputVolume">2500</ field> </fields> </entity> </updateField> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with deleteField request to delete the inputVolume and the outputVolume fields in the Q1 Quota row in the quota entity. The Two rows are called Q1 exist, one row with a CID of 111 and another row with a CID of 222. The request is to delete the instance with the CID of 111.	<pre> <deleteField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>Quota</name>Non Transaction Request with updateField request to <interface>XMLIMPORT</interface> <xpath>/usage/quota[@name='Q1' and cid='111']</xpath> </data> <fields> <field name="inputVolume">3000</ field> <field name="outputVolume">2500</ field> </fields> </entity> </deleteField> </pre>
Non Transaction Request with updateField request to update the InitialInputVolume and the InitialOutputVolume fields in the DQ1DynamicQuota row in the DynamicQuota entity. The two rows are called DQ1 exist, one row with an InstanceId of 11111 and another row with an InstanceId of 22222. The request is to update the instance with the InstanceId of 11111	<pre> <updateField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath>/definition/ DynamicQuota[@name='DQ1' and InstanceId='11111']</xpath> </data> <fields> <field name="InitialInputVolume">3000</ field> <field name="InitialOutputVolume">2500</ field> </fields> </entity> </updateField> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with deleteField request to delete the InitialInputVolume and the InitialOutputVolume fields in the DQ1DynamicQuota row in the DynamicQuota entity. The two rows are called DQ1 exist, one row with an InstanceId of 11111 and another row with an InstanceId of 22222. The request is to delete the instance with the InstanceId of 11111	<pre> <deleteField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>DynamicQuota</name> <interface>XMLIMPORT</interface> <xpath>/definition/ DynamicQuota[@name='DQ1' and InstanceId='11111']</xpath> </data> <fields> <field name="InitialInputVolume">3000</ field> <field name="InitialOutputVolume">2500</ field> </fields> </entity> </deleteField> </pre>
Non transaction request with updateField request to update a property value in the State entity. The two properties exist, one with a propertyname of MCC and another with name of approved. The request is to update the property value with the name of MCC.	<pre> <updateField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>State</name> <interface>XMLIMPORT</interface> <xpath>/state/property[name='mcc']</ xpath> </data> <fields> <field name="value">3000</field> </fields> </entity> </updateField> </pre>

Table 2-49 (Cont.) iXML Format for Bulk Import

Opertion Detail	iXML Format
Non transaction request with deleteField request to delete a property value in the State entity. The two properties exist, one with a propertyname of MCC and another with name of approved. The request is to delete the property value with the name of MCC.	<pre> <deleteField clearAll="true"> <key> <MSISDN>15141234567</MSISDN> </key> <entity> <data> <name>State</name> <interface>XMLIMPORT</interface> <xpath>/state/property[name='mcc']</xpath> </data> <fields> <field name="value">3000</field> </fields> </entity> </deleteField> </pre>

2.37.1.3 Bulk Import Operations in CSV Format

The below table provides operation details of bulk import tool in CSV format.

Table 2-50 CSV Format for Bulk Import

Opertion Detail	CSV
Transaction block with single request of Entity Dynamic Quota (update request)	<pre> keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,447590800589,MODIFY,PCF- VSA,{"definition":{"DynamicQuota": [{"Type":"pass","InstanceId":"7481519 47_2021-01-25T21.02.31_00001","Priori ty":"1","expirationdatetime":"2021-02 -24T23:59:59","InitialTotalVolume":"8 47447040","name":"SMB_TOP","purchased atetime":"2021-01-25T21:02:31"}]],"ver sion":3}}", </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
Transaction block with uploadField and deleteField requests	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,1000000017,PATCH,PCF- VSA,"[{\"op\":\"add\",\"patchFields\": {\"IMSI\": [\"302370111111111\"]},\"filters\":null}, {\"op\":\"add\",\"patchFields\": {\"BillingDay\":23},\"filters\":null}, {\"op\":\"delete\",\"patchFields\":{\"IMSI\": [\"302370111111121\"]},\"filters\":null}]] \",/subscriber
Transaction block with multiple requests of updateField , deleteField , and deleteFieldSet of State entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF- VSA,"[{\"op\":\"add\",\"patchFields\": {\"mcc\":\"3000\"},\"filters\":null}, {\"op\":\"delete\",\"patchFields\": {\"expire\":\"2010-02-09T11:20:32\"},\"fil ters\":null}, {\"op\":\"delete\",\"patchFields\": {\"approved\":null},\"filters\":null}]]\",/ state/property/row

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
Transaction block request with all entities (createSubscriber along with Create request for Quota, DynamicQuota, and State)	<pre> keyType, keyValu, operationType, nfType , jsonPayload, addInfo MSISDN, 33123654862, CREATE, PCF- VSA, {"subscriber":{"MSISDN": ["33123654862"], "IMSI": ["184569547984229"], "ACCOUNTID": ["10404723525"], "BillingDay":1, "Entit lement": ["DayPass", "DayPassPlus"]}, "usage": {"quota": [{"totalVolume":"1000", "RefInstanceId ":""," grantedTotalVolume":"0", "output Volume":"20", "Type":""," grantedInputV olume":"0", "grantedOutputVolume":"0", "grantedTime":""," grantedServiceSpeci fic":""," name":"AggregateLimit", "next ResetTime":"2011-04-22T00:00:00-05:00 ", "time":"3422", "inputVolume":"980", serviceSpecific":"12", "cid":"92233720 36854775807", "QuotaState":""}], "versi on":"3"}, "definition": {"version":"1", "dynamicquota": [{"InstanceId":"15678", "Initialservic eSpecific":"4", "ActivationDateTime":" 2015-04-22T00:00:00-05:00", "Priority" :"4", "ExpirationDateTime":"2015-06-22 T00:00:00-05:00", "InterimReportingInt erval":"100", "Duration":"10", "Type":" Roll- Over", "InitialoutputVolume":"500", "na me":"AggregateLimit", "InitialTime":"1 35", "InitialtotalVolume":"2000", "Init ialinputVolume":"1500"}]}, "state": {"property": [{"name":"mcc", "value":"315"}, {"name":"expire", "value":"2010-02-09T 11:20:32"}, {"name":"approved", "value":"yes"}], "v ersion":"1"}} </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
Transaction block request with all Entities (createSubscriber along with updateFieldSet request for Quota, DynamicQuota, and State)	<pre> keyType, keyvalue, operationType, nfType , jsonPayload, addInfo MSISDN, 33123654862, CREATE, PCF- VSA, {"subscriber":{"MSISDN": ["33123654862"], "IMSI": ["184569547984229"], "ACCOUNTID": ["10404723525"], "BillingDay":1, "Entit lement": ["DayPass", "DayPassPlus"]}, "usage": {"quota": [{"totalVolume":"1000", "RefInstanceId ":""," grantedTotalVolume":"0", "output Volume":"20", "Type":""," grantedInputV olume":"0", "grantedOutputVolume":"0", "grantedTime":""," grantedServiceSpeci fic":""," name":"AggregateLimit", "next ResetTime":"2011-04-22T00:00:00-05:00 ", "time":"3422", "inputVolume":"980", " serviceSpecific":"12", "cid":"92233720 36854775807", "QuotaState":""}], "versi on":"3"}, "definition": {"version":"1", "dynamicquota": [{"InstanceId":"15678", "Initialservic eSpecific":"4", "ActivationDateTime":" 2015-04-22T00:00:00-05:00", "Priority" :"4", "ExpirationDateTime":"2015-06-22 T00:00:00-05:00", "InterimReportingInt erval":"100", "Duration":"10", "Type":" Roll- Over", "InitialoutputVolume":"500", "na me":"AggregateLimit", "InitialTime":"1 35", "InitialtotalVolume":"2000", "Init ialinputVolume":"1500"}]}, "state": {"property": [{"name":"mcc", "value":"315"}, {"name":"expire", "value":"2010-02-09T 11:20:32"}, {"name":"approved", "value":"yes"}], "v ersion":"1"}}, </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
Non transaction request to Create Subscriber	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo AccountId,10404723525,CREATE,PCF- VSA,"{"subscriber":{"MSISDN": ["33123654862"],"Tier":"","IMSI": ["184569547984229"],"ACCOUNTID": ["10404723525"],"BillingDay":1,"Entit lement":["DayPass","DayPassPlus"]}}",
Non transaction request to Update Subscriber	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,33123654862,UPDATE,PCF- VSA,"{"subscriber": {"AccountId":"10404723525","MSISDN": ["33123654862"],"Tier":"Silver","IMSI ": ["184569547984229"],"BillingDay":6,"E ntitlement":["DayPass"]}}",
Non transaction request to Delete Subscriber	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,33123654862,DELETE,PCF-VSA,,
Non transaction request with Create request to create Quota Entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,MODIFY,PCF- VSA,"{"usage":{"quota": [{"totalVolume":"1000","name":"Aggreg ateLimit","nextResetTime":"2010-05-22 T00:00:00-05:00","time":"3422","outpu tVolume":"20","inputVolume":"980","se rviceSpecific":"12","cid":"9223372036 854775807"}],"version":3}}",

Table 2-50 (Cont.) CSV Format for Bulk Import

Operation Detail	CSV
Non transaction request with Create request to create State Entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,MODIFY,PCF- VSA,{"state":{"property": [{"name":"mcc","value":"315"}, {"name":"expire","value":"2014-02-09T 11:20:32"}, {"name":"approved","value":"no"}],"ve rsion":1}}},
Non transaction request with Create request to create Dynamic Quota Entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,MODIFY,PCF- VSA,{"definition":{"DynamicQuota": [{"InstanceId":"15678","Priority":"4" ,"InitialInputVolume":"1500","Interim ReportingInterval":"100","Duration":" 10","InitialOutputVolume":"500","Type ":"Roll- over","activationdatetime":"2015-03-0 9T11:20:32","expirationdatetime":"201 5-04-09T11:20:32","InitialTotalVolume" :"2000","name":"PDQ1","InitialService Specific":"4","InitialTime":"135"}]," version":3}}},
Non transaction request with UpdateFieldSet request to update Quota Entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,MODIFY,PCF- VSA,{"usage":{"quota": [{"totalVolume":"1000","name":"Aggreg ateLimit","nextResetTime":"2010-05-22 T00:00:00-05:00","time":"3422","outpu tVolume":"20","inputVolume":"980","se rviceSpecific":"12","cid":"9223372036 854775807"}],"version":3}}},

Table 2-50 (Cont.) CSV Format for Bulk Import

Operation Detail	CSV
Non transaction request with UpdateFieldSet request to update State Entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,MODIFY,PCF-VSA,{"state":{"property":[{"name":"mcc","value":"315"}, {"name":"expire","value":"2014-02-09T11:20:32"}, {"name":"approved","value":"no"}],"version":1}}},
Non transaction request with UpdateFieldSet request to update DynamicQuota entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,MODIFY,PCF-VSA,{"definition":{"DynamicQuota":[{"InstanceId":"15678","Priority":"4","InitialInputVolume":"1500","InterimReportingInterval":"100","Duration":"10","InitialOutputVolume":"500","Type":"Roll-over","activationdatetime":"2015-03-09T11:20:32","expirationdatetime":"2015-04-09T11:20:32","InitialTotalVolume":"2000","name":"PDQ1","InitialServiceSpecific":"4","InitialTime":"135"}],"version":3}}},
Non transaction request with deleteFieldSet request to delete Quota entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,DELETE,PCF-VSA,,?fields=usage
Non transaction request with deleteFieldSet request for Quota to delete row Q2	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,DELETE,PCF-VSA,,/usage/quota/row?name=Q2
Non transaction request with deleteFieldSet request to delete State entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,DELETE,PCF-VSA,,?fields=state

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
Non transaction request with deleteFieldSet request for State entity to delete field MCC	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF-VSA,"[{\"op\":\"delete\",\"patchFields\":{\"mcc\":null},\"filters\":null}]",/state/property/row
Non transaction request with deleteFieldSet request to delete Dynamic Quota Entity	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,DELETE,PCF-VSA,,?fields=definition
Non transaction request with deleteFieldSet request for Dynamic Quota to delete row PDQ1	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,DELETE,PCF-VSA,,/definition/DynamicQuota/row?name=PDQ1
Non transaction request with updateField request to update a subscriber profile and set the value of the BillingDay field to 23 and the Tier field to Gold.	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF-VSA,"[{\"op\":\"replace\",\"patchFields\":{\"BillingDay\":23},\"filters\":null},{\"op\":\"replace\",\"patchFields\":{\"Tier\":\"Gold\"},\"filters\":null}]",/subscriber

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
Non transaction request with updateField request to update a subscriber profile. The request performs the following functions: - Replaces all exist IMSI values with a single value - Deletes the specific MSISDN 14161112222 - Deletes all instances of the NAI - uupdates the BillingDay field - Deletes the Tier field if the existing value is set to Gold - Deletes the Custom5 field	<pre>keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF- VSA,"[{\"op\":\"add\",\"patchFields\": {\"IMSI\": [\"302370111111111\"],\"filters\":null}, {\"op\":\"delete\",\"patchFields\": {\"MSISDN\": [\"14161112222\"],\"filters\":null}, {\"op\":\"delete\",\"patchFields\": {\"NAI\":null,\"filters\":null}, {\"op\":\"add\",\"patchFields\": {\"BillingDay\":11,\"filters\":null}, {\"op\":\"delete\",\"patchFields\": {\"Tier\":\"Gold\"},\"filters\":null}, {\"op\":\"delete\",\"patchFields\": {\"Custom5\":null,\"filters\":null}}]\",/ subscriber</pre>
Non transaction request with deleteField request to delete a subscriber profile and set the value of the BillingDay field to 23 and the Tier field to Gold	<pre>keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF- VSA,"[{\"op\":\"delete\",\"patchFields\": {\"BillingDay\":23,\"filters\":null}, {\"op\":\"delete\",\"patchFields\": {\"Tier\":\"Gold\"},\"filters\":null}}]\",/ subscriber</pre>
Non transaction request with updateField request to update the inputVolume and the outputVolume fields in the Q1 Quota row in the Quota entity. The two rows are called Q1 exist, one row with a CID of 111 and another row with a CID of 222. The request is to update the instance with the CID of 111.	<pre>keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF- VSA,"[{\"op\":\"replace\",\"patchFields\": {\"inputVolume\":\"3000\"},\"filters\": {\"name\":\"Q1\",\"cid\":\"111\"}}, {\"op\":\"replace\",\"patchFields\": {\"outputVolume\":\"2500\"},\"filters\": {\"name\":\"Q1\",\"cid\":\"111\"}}]\",/usage/ quota/row</pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Operation Detail	CSV
Non transaction request with deleteField request to delete the inputVolume and the outputVolume fields in the Q1 Quota row in the quota entity. The Two rows are called Q1 exist, one row with a CID of 111 and another row with a CID of 222. The request is to delete the instance with the CID of 111.	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF-VSA,"[{\"op\":\"delete\",\"patchFields\":{\"inputVolume\":\"3000\"},\"filters\":{\"name\":\"Q1\",\"cid\":\"111\"}}},{\"op\":\"delete\",\"patchFields\":{\"outputVolume\":\"2500\"},\"filters\":{\"name\":\"Q1\",\"cid\":\"111\"}}]\",/usage/quota/row
Non Transaction Request with updateField request to update the InitialInputVolume and the InitialOutputVolume fields in the DQ1DynamicQuota row in the DynamicQuota entity. The two rows are called DQ1 exist, one row with an InstanceId of 11111 and another row with an InstanceId of 22222. The request is to update the instance with the InstanceId of 11111	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF-VSA,"[{\"op\":\"replace\",\"patchFields\":{\"InitialInputVolume\":\"3000\"},\"filters\":{\"InstanceId\":\"11111\",\"name\":\"DQ1\"}}},{\"op\":\"replace\",\"patchFields\":{\"InitialOutputVolume\":\"2500\"},\"filters\":{\"InstanceId\":\"11111\",\"name\":\"DQ1\"}}]\",/definition/DynamicQuota/row
Non transaction request with deleteField request to delete the InitialInputVolume and the InitialOutputVolume fields in the DQ1DynamicQuota row in the DynamicQuota entity. The two rows are called DQ1 exist, one row with an InstanceId of 11111 and another row with an InstanceId of 22222. The request is to delete the instance with the InstanceId of 11111	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF-VSA,"[{\"op\":\"delete\",\"patchFields\":{\"InitialInputVolume\":\"3000\"},\"filters\":{\"InstanceId\":\"11111\",\"name\":\"DQ1\"}}},{\"op\":\"delete\",\"patchFields\":{\"InitialOutputVolume\":\"2500\"},\"filters\":{\"InstanceId\":\"11111\",\"name\":\"DQ1\"}}]\",/definition/DynamicQuota/row
Non transaction request with updateField request to update a property value in the State entity. The two properties exist, one with a propertyname of MCC and another with name of approved. The request is to update the property value with the name of MCC.	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF-VSA,"[{\"op\":\"replace\",\"patchFields\":{\"mcc\":\"3000\"},\"filters\":null}]",/state/property/row

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
Non transaction request with deleteField request to delete a property value in the State entity. The two properties exist, one with a propertyname of MCC and another with name of approved. The request is to delete the property value with the name of MCC.	keyType,keyValue,operationType,nfType ,jsonPayload,addInfo MSISDN,15141234567,PATCH,PCF- VSA,"[{\"op\":\"delete\",\"patchFields\": {\"mcc\":\"3000\"},\"filters\":null}]",/ state/property/row

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
5G PCF Data includes complete policy data information	<pre> keyType,keyValue,operationType,nfType ,jsonPayload imsi,1111111111,MODIFY,PCF, "{ "am-data": { "praInfos": { "p1": { "praId": "p1", "trackingAreaList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "tac": "5CB7" }], "ecgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, }, "eutraCellId": "92FFdBE" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" }], </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "ncgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" }], "globalRanNodeIdList": [{ "plmnId": { "mcc": "965", "mnc": "235" }, "n3IwfId": "FFf0f2AFbFa16CEfE7" }, { "plmnId": { "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25, "gNbValue": "1A8F1D" } }], }, </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "subscCats": ["cat1", "cat2"], "sm-data": { "umData": { "mk1": { "scopes": { "11-abc123": { "dnn": ["dnn1"], "snssai": { "sd": "abc123", "sst": 11 } } }, "limitId": "mk1", "umLevel": "SERVICE_LEVEL", "resetTime": "2018-01-02T08:17:14.090Z", "allowedUsage": { "duration": 9000, "totalVolume": 8888, "uplinkVolume": 6666, "downlinkVolume": 7777 } }, "umDataLimits": { "mk1": { "scopes": { "11-abc123": { "dnn": ["dnn1"], "snssai": { "sd": "abc123", "sst": 11 } } } } } } </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> }, "endDate": "2018-11-05T08:17:14.090Z", "limitId": "mk1", "umLevel": "SESSION_LEVEL", "startDate": "2018-09-05T08:17:14.090Z", "usageLimit": { "duration": 6000, "totalVolume": 9000, "uplinkVolume": 5000, "downlinkVolume": 4000 }, "resetPeriod": { "period": "YEARLY" } }, "smPolicySnssaiData": { "11-abc123": { "snssai": { "sd": "abc123", "sst": 11 }, "smPolicyDnnData": { "dnn1": { "dnn": "dnn1", "bdtRefIds": { "xyz": "bdtRefIds", "abc": "xyz" }, "gbrDl": "7788 Kbps", "gbrUl": "5566 Kbps", "online": true, "chfInfo": { </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "primaryChfAddress": "1.1.1.1", "secondaryChfAddress": "2.2.2.2" }, "offline": true, "praInfos": { "p1": { "praId": "p1", "trackingAreaList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "tac": "5CB7" }], "ecgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }] </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> } , { "plmnId": { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" }, "ncgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" }, , { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" },], "globalRanNodeIdList": [{ "plmnId": { "mcc": "965", </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre>"mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" } , { "plmnId": { "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25, "gNbValue": "1A8F1D" } }] }, "ipv4Index": 0, "ipv6Index": 0, "subscCats": ["cat1", "cat2"], "adcSupport": true, "mpsPriority": true, "allowedServices": ["ser1", "ser2"],</pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "mpsPriorityLevel": 2, "imsSignallingPrio": true, "refUmDataLimitIds": { "mk1": { "monkey": ["monkey1"], "limitId": "mk1" }, }, "subscSpendingLimits": true } } }, }, "ue-policy-set": { "praInfos": { "p1": { "praId": "p1", "trackingAreaList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "tac": "5CB7" }], "ecgiList": [{ "plmnId": { "mcc": </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" }], "ncgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" }], "globalRanNodeIdList": [{ "plmnId": { "mcc": "965", "mnc": "235" }, "n3IwfId": "FFf0f2AFbFa16CEfE7" </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> }, { "plmnId": { "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25, "gNbValue": "1A8F1D" } }] } }, "subscCats": ["cat1", "cat2"], "uePolicySections": { "33": { "upsi": "33", "uePolicySectionInfo": "MQ==" }, "111": { "upsi": "111", "uePolicySectionInfo": "Mg0K" } }, "upsis": ["111", "33"], "allowedRouteSelDescs": { "965-235": { "servingPlmn": { "mcc": "965", "mnc": "235" } }, "snssaiRouteSelDescs": [{ "snssai": { "sd": "abc123", </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "sst": 11 }, "dnnRouteSelDescs": [{ "dnn": "dnn1", "sscModes": ["SSC_MODE_1", "SSC_MODE_2"], "pduSessTypes": ["IPv4"] }] } }, "andspInd": true, "pei": "imei-355375345274757", "osIds": ["bceafb30-31bd-4b8d-b3c2- ed79d029531a", "cc9d9f97-0127-497c-9bca-79e8e9080ea8 "] } }"</pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
5G and 4G PCF Data includes complete policy data information along with VSA data	<pre> keyType,keyValue,operationType,nfType ,jsonPayload imsi,1111111111,MODIFY,PCF, "{ "am-data": { "praInfos": { "p1": { "praId": "p1", "trackingAreaList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "tac": "5CB7" }], "ecgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" }], </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "ncgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" }], "globalRanNodeIdList": [{ "plmnId": { "mcc": "965", "mnc": "235" }, "n3IwfId": "FFf0f2AFbFa16CEfE7" }, { "plmnId": { "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25, "gNbValue": "1A8F1D" } }], }, </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "subscCats": ["cat1", "cat2"], "sm-data": { "umData": { "mk1": { "scopes": { "11-abc123": { "dnn": ["dnn1"], "snssai": { "sd": "abc123", "sst": 11 } } }, "limitId": "mk1", "umLevel": "SERVICE_LEVEL", "resetTime": "2018-01-02T08:17:14.090Z", "allowedUsage": { "duration": 9000, "totalVolume": 8888, "uplinkVolume": 6666, "downlinkVolume": 7777 } }, "umDataLimits": { "mk1": { "scopes": { "11-abc123": { "dnn": ["dnn1"], "snssai": { "sd": "abc123", "sst": 11 } } } } } } </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> }, "endDate": "2018-11-05T08:17:14.090Z", "limitId": "mk1", "umLevel": "SESSION_LEVEL", "startDate": "2018-09-05T08:17:14.090Z", "usageLimit": { "duration": 6000, "totalVolume": 9000, "uplinkVolume": 5000, "downlinkVolume": 4000 }, "resetPeriod": { "period": "YEARLY" } }, "smPolicySnssaiData": { "11-abc123": { "snssai": { "sd": "abc123", "sst": 11 }, "smPolicyDnnData": { "dnn1": { "dnn": "dnn1", "bdtRefIds": { "xyz": "bdtRefIds", "abc": "xyz" }, "gbrDl": "7788 Kbps", "gbrUl": "5566 Kbps", "online": true, "chfInfo": { "primaryChfAddress": "1.1.1.1", </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "secondaryChfAddress": "2.2.2.2" }, "offline": true, "praInfos": { "p1": { "praId": "p1", "trackingAreaList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "tac": "5CB7" }], "ecgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }] </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre>, { "plmnId": { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" }, "ncgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" }, , { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" },], "globalRanNodeIdList": [{ "plmnId": { "mcc": "965", "mnc": "235"</pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> }, "n3IwfId": "fFf0f2AFbFa16CEfE7" } , { "plmnId": { "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25, "gNbValue": "1A8F1D" } }] }, "ipv4Index": 0, "ipv6Index": 0, "subscCats": ["cat1", "cat2"], "adcSupport": true, "mpsPriority": true, "allowedServices": ["ser1", "ser2"], "mpsPriorityLevel": 2, "imsSignallingPrio": true,</pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "refUmDataLimitIds": { "mk1": { "monkey": ["monkey1"], "limitId": "mk1" }, "subscSpendingLimits": true }, "VendorSpecific-000111": { "subscriber": { "IMSI": ["302370123456789"], "ACCOUNTID": ["178322212122"], "MSISDN": ["15145551234", "15141234567"], "BillingDay": 6, "Entitlement": ["DayPass", "DayPassPlus"] }, "state": { "property": [{ "name": "mcc", "value": "123" }, { "name": "expire", "value": "201402-09T11:20:32" }], "version": 1 } }, "ue-policy-set": { "praInfos": { "p1": { "praId": "p1", "trackingAreaList": [{" </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "plmnId": { "mcc": "976", "mnc": "32" }, "tac": "5CB6" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "tac": "5CB7" }], "ecgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, { "plmnId": { "mcc": "977", "mnc": "33" }, "eutraCellId": "8F868C4" }], "ncgiList": [{ "plmnId": { "mcc": "976", "mnc": "32" }, "nrCellId": "b2fB6fE9D" }] </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> { "plmnId": { "mcc": "977", "mnc": "33" }, "nrCellId": "5d1B4127b" }, "globalRanNodeIdList": [{ "plmnId": { "mcc": "965", "mnc": "235" }, "n3IwfId": "fFf0f2AFbFa16CEfE7" }, { "plmnId": { "mcc": "967", "mnc": "238" }, "gNbId": { "bitLength": 25, "gNbValue": "1A8F1D" } }], "subscCats": ["cat1", "cat2"], "uePolicySections": { "33": { "upsi": "33", "uePolicySectionInfo": "MQ==" }, </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Opertion Detail	CSV
	<pre> "111": { "upsi": "111", "uePolicySectionInfo": "Mg0K" } }, "upsis": ["111", "33"], "allowedRouteSelDescs": { "965-235": { "servingPlmn": { "mcc": "965", "mnc": "235" }, "snsaiRouteSelDescs": [{ "snsai": { "sd": "abc123", "sst": 11 }, "dnnRouteSelDescs": [{ "dnn": "dnn1", "sscModes": ["SSC_MODE_1", "SSC_MODE_2"], "pduSessTypes": ["IPv4"] }] }, "andspInd": true, "pei": "imei-355375345274757", "osIds": ["bceafb30-31bd-4b8d-b3c2- ed79d029531a", "cc9d9f97-0127-497c-9bca-79e8e9080ea8 </pre>

Table 2-50 (Cont.) CSV Format for Bulk Import

Operation Detail	CSV
	<pre>"] } }"</pre>

Multiple Pod Subscriber Bulk Import Support

Update the below configuration as required to support subscriber bulk import tool for multiple pod.

```
deployment:
  # Replica count for deployment
  replicaCount: 1
```

The number of subscriber bulk import pod is as per the given replicaCount. Pod name must be used when copying the CSV, XML, or PDBI files.

```
kubectl cp <csv file> <namespace>/<nudr-bulk-import pod name>:/home/udruser/
import
```

For example, there are three replicas each have its own pod and PVC.

```
kubectl cp create_pcf.csv bulktest/bulktest-nudr-bulk-import-0:/home/udruser/
import
```

```
kubectl cp create_pcf.csv bulktest/bulktest-nudr-bulk-import-1:/home/
udruser/xml
```

```
kubectl cp create_pcf.csv bulktest/bulktest-nudr-bulk-import-2:/home/udruser/
pdbi
```

When files are copied to particular pod, it will process and generate the result logs files to its own PVC.

Note

- SFTP do not support multiple pod bulk import.
- Copying files into or from PVC must be within the pod.

2.37.2 Installing Subscriber Bulk Import Tool

You can install the subscriber bulk import and xmldata containers using helm. While installing the bulk import and xmldata containers, the "bulkimportpersistentclaim" PVC for bulk import container and the "xmldatapersistentclaim" PVC for xmldata container are created automatically. This is because the tool uses Kubernetes StatefulSet for deployment and it has a limitation to create PVC when the tool is installed.

Depending on the resource availability to get better performance of the subscriber bulk import tool change the `maxConnectionsPerDestination` parameter value range between 4 to 200 in the `custom-value.yaml` file.

```
#Jetty client config
maxRequestsQueuedPerDestination: 20000
maxConnectionsPerDestination: 4
requestTimeout: 5000 #(ms)
jettyMaxPoolSize: 1000
jettyMinPoolSize: 50
jettyIdleTimeout: 0 #(ms,<=0 -> to make timeout infinite)
```

Installing Subscriber Bulk Import Tool

Before installing the bulk import tool using helm charts, you are required to configure its parameters in the `nudr-bulk-import-custom-values.yaml` file. For information about the bulk import tool configurable parameters, see [Configuring Subscriber Bulk Import Tool](#).

To install the bulk import tool as a separate deployment with different release name:

1. Extract the bulk import helm charts from the software or CSAR package and run the following command:
`helm3 install <release name for bulk import> -n <udr namespace> nudr-bulk-import/ -f nudr-bulk-import-custom-values.yaml`

Note

Run the following command to upgrade the bulk import microservice:
`helm3 upgrade <release name for bulk import> -n <udr namespace> -f nudr-bulk-import-custom-values.yaml`

2. Once the pod is up and running, copy the CSV file. The container starts processing the records.

Figure 2-32 Bulk Import Install

```
Every 2.0s: kubectl get pods -n udr
```

NAME	READY	STATUS	RESTARTS	AGE
bulk-nudr-bulk-import-0	1/1	Running	0	2m8s
ocudr-appinfo-759cd598c4-kgbbn	1/1	Running	0	20m
ocudr-egressgateway-bb846c898-2vg6n	1/1	Running	0	20m
ocudr-ingressgateway-dbb9b84b-cnmlr	1/1	Running	0	20m
ocudr-nudr-config-8578bcd968-z5tbk	1/1	Running	0	20m
ocudr-nudr-config-server-689b96bc6c-c75hk	1/1	Running	0	20m
ocudr-nudr-drsservice-74cf4bcbf6-bndf6	1/1	Running	0	20m
ocudr-nudr-nrf-client-nfmanagement-77b5d67648-dfcmc	1/1	Running	0	20m
ocudr-oc-diam-gateway-0	1/1	Running	0	20m

Verifying Subscriber Bulk Import PVC

To verify the bulk import PVC:

1. Run the following command to check PVC creation:
`kubectl get pvc -n <namespace>`

Figure 2-33 Verifying PVC

```
[cloud-user@cne-180-dev2-bastion-1 xmltocsv]$ kubectl get pvc -n udr;
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
bulkimportpersistentclaim-bulk-nudr-bulk-import-0	Bound	pvc-067419cb-cc18-4780-944d-bcd9958c1d3b	10Gi	RWO	standard	3m29s

OR

Run the following command to verify the PV creation:

```
kubectl get pv
```

Figure 2-34 Verifying PV

pvc-067419cb-cc18-4780-944d-bcd9958c1d3b	10Gi	RWO	Delete	Bound	udr/bulkimportpersistentclaim-bulk-nudr-bulk-import-0

2. Update the values.yaml file available at `ocudr/charts/nudr-bulk-import` and run the following command to upgrade the bulk import tool.

```
helm3 upgrade <release name for bulk import> -n <udr namespace> .;
```

Installing xmltocsv converter

To install xmltocsv container:

1. Enable the following flag in the custom-values.yaml file. The custom-values.yaml file is available at `/nudr-bulk-import` location.

```
nudr-bulk-import:
  xmlToCsv:
    enabled: true(default value is false)
```

2. Run the following command to install the bulk import tool:
`helm3 install <release name for bulk import> -n <udr namespace> -f nudr-bulk-import-custom-values.yaml`

- Once the pod is up and running, copy the ixml file. The xmltocsv container starts processing the records.

Figure 2-35 xmltocsv install

```
Every 2.0s: kubectl get pods -n udr
```

NAME	READY	STATUS	RESTARTS	AGE
bulk-nudr-bulk-import-0	2/2	Running	0	9m35s
ocudr-appinfo-759cd598c4-6f8zs	1/1	Running	0	13m
ocudr-egressgateway-bb846c898-k9mlr	1/1	Running	0	13m
ocudr-ingressgateway-dbb9b84b-ndd2j	1/1	Running	0	13m
ocudr-nudr-config-8578bcd968-4lglr	1/1	Running	0	13m
ocudr-nudr-config-server-689b96bc6c-7wktr	1/1	Running	0	13m
ocudr-nudr-drservice-74cf4bcbf6-nk7dk	1/1	Running	0	13m
ocudr-nudr-nrf-client-nfmanagement-77b5d67648-tr8xt	1/1	Running	0	13m
ocudr-oc-diam-gateway-0	1/1	Running	0	13m

Verifying xmltocsv PVC Creation

To verify bulk import and xmltocsv PVC creation:

- Run the following command to verify bulk import and xmltocsv PVC creation:
`kubectl get pvc -n <namespace>`

Figure 2-36 Verifying xmltocsv pvc

```
[cloud-user@cne-180-dev2-bastion-1 ocudr]$ kubectl get pvc -n udr
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
bulkimportpersistentclaim-bulk-nudr-bulk-import-0	Bound	pvc-067419cb-cc18-4780-944d-bcd9958c1d3b	10Gi	RWO	standard	22m
xmltocsvpersistentclaim-bulk-nudr-bulk-import-0	Bound	pvc-d4986682-7c39-45c1-b984-f85722344f02	10Gi	RWO	standard	10m

- Run the following command to verify the PV creation:
`kubectl get pv`

Figure 2-37 Verifying xmltocsv pv

pvc-067419cb-cc18-4780-944d-bcd9958c1d3b	10Gi	RWO	Delete	Bound	udr/bulkimportpersistentclaim-bulk-nudr-bulk-import-0
pvc-d4986682-7c39-45c1-b984-f85722344f02	10Gi	RWO	Delete	Bound	udr/xmltocsvpersistentclaim-bulk-nudr-bulk-import-0

① Note

- If the bulk import tool is already running on your system, then you cannot upgrade to add xmltocsv container to an existing bulk import container. Uninstall the existing bulk import container and enable xmlToCsv flag by running the following command:
`helm3 install <release name for bulk import> -n <udr namespace> -f nudr-bulk-import-custom-values.yaml`
- If both the containers are already running, then update the custom-values.yaml file in the at /nudr-bulk-import location and run the following command to upgrade bulk import tool.

```
helm3 upgrade <release name for bulk import> -n <udr namespace> -f nudr-bulk-import-custom-values.yaml
```

To Install the Subscriber Bulk Import Tool for EIR Provisioning

1. Enable the following flag in the nudr-bulk-import-custom-values.yaml file.

```
pdbiToCsv:
  enabled: true
```

2. Configure the PDBI file path, where the PDBI nudr-bulk-import-custom-values.yaml file are replaced, using the following configuration.

```
createPdbiToCsvPVC:
  pathLocation: /home/udruser/pdbi
```

3. Run the following command to install the bulk import tool for EIR provisioning.

```
helm install <release-Name> -n <namespace> -f <nudr-bulk-import-
customvalues.yaml file> nudr-bulk-import-23.1.2.tgz
```

The bulk import will start running with 2 container. The container starts processing the records.

pdbiimport-nudr-bulk-import-0	3/3	Running	0
udr1-appinfo-67fff6d8f8-rkkkn	1/1	Running	0
udr1-egressgateway-5bf94b5bdf-j4wkk	1/1	Running	0
udr1-ingressgateway-prov-c8c97f596-hcgms	1/1	Running	0
udr1-ingressgateway-sig-5468cc6984-k94rk	1/1	Running	0
udr1-nudr-config-6d456d8bb7-9zr5n	1/1	Running	0
udr1-nudr-config-server-7cf7549f64-9tbrk	1/1	Running	0
udr1-nudr-diameterproxy-775db94f46-s9x4v	1/1	Running	0
udr1-nudr-dr-provservice-58d4cdcc6d-h567v	1/1	Running	0
udr1-nudr-dr-service-6989c6b48c-clrtr	1/1	Running	0

Running and Monitoring the Bulk Import Tool for EIR

1. Place the PDBI files in the configured PDBI file path location /home/udruser/pdbi folder.
2. The PDBI file is converted to CSV file and is placed in /home/udruser/import folder.
3. The bulk import main container processes the CSV file where the provisioning API's for EIR are called and the log files are also generated.

Sample Logs Files:

Containers generate summary report log file and the final result log file in configured PDBI file path/logs. Below are the samples of the log files:

Sample log file after processing PDBI file to CSV file:

```

bash-4.4$ cat pdbi_tool_create1.log
PDBI to CSV Tool Summary Report :
Time taken for processing file : 0m6s
Number Of Failed Record :0
Total Number of records fetched from pdbi file :1000000
Total Number of valid records :1000000
Total Number of invalid records :0
Total Number success for create operation :0
Total Number of failures for create operation :0
Total Number success for modify operation :1000000
Total Number of failures for modify operation :0
Total Number success for delete operation :0
Total Number of failures for delete operation :0bash-4.4$

```

Sample log file after processing CSV file:

```

Import Tool Summary Report :
Time taken for processing file : 6m36s
Number Of Failed Record :0
Total Number of records fetched from csv files :1000000
Total Number of invalid records :0
Total Number of valid records :1000000
Total Number success for create operation :0
Total Number of failures for create operation :0
Total Number success for modify operation :1000000
Total Number of failures for modify operation :0
Total Number success for delete operation :0
Total Number of failures for delete operation :0
Total Number of failures for service time out :0
List of failed keys :
All Records Successbash-4.4$

```

Sample PDBI result log file: PDBI result log file is stored in the configured PDBI file path/logs (example: /home/udruser/pdbi/logs).

```

ent_eir(iid 8904,imei 12312312008904,allow no,gray yes,block yes)
rc=PDBI_INVALID_VALUE
ent_eir(iid 9336,imei 12312312009336,allow no,gray yes,block yes)
rc=PDBI_INVALID_VALUE
PDBI Import Tool Summary Report :
Time taken for processing file : 10s
Total Number of records fetched from PDBI files : 10000
Successful: 0 Failures: 10000

```

Change the Configuration for an Existing Subscriber Bulk Import Tool using Helm

Perform the following steps to change the configuration for an existing Subscriber Bulk Import Tool deployment:

Note

This configuration is not exposed through the REST API.

1. Modify the Subscriber Bulk Import Tool custom yaml file with the required configuration.
2. Run the following command to delete the existing subscriber bulk import tool Helm deployment

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

3. Run the following command to install the subscriber bulk import tool with the modified custom yaml file

```
helm install <release name> -n <name space> <chart path> -f <values.yaml  
path with latest configuration>
```

2.37.2.1 Configuring Subscriber Bulk Import Tool

The following table lists the user parameters for creating PVC and template for installing Subscriber Bulk Import Tool.

Table 2-51 Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.dockerRegistry	Docker registry from where the images are pulled	ocudr-registry.us.abc.com:5000/ocudr	NA	NA
global.mysql.dbServiceName	Database service to connect	mysql-connectivity-service.occne-infra	NA	NA
global.mysql.port	Port for database service connection	3306	NA	NA
global.udrTracing.enable	Flag to enable UDR tracing on Jaeger	false	true or false	NA
global.udrTracing.host	Jaeger service name installed in CNE	occne-tracer-jaeger-agent.occne-infra	NA	NA
global.udrTracing.port	Jaeger Service Port installed in CNE	6831	NA	NA
global.serviceAccountName	The entire pod should be associated with the service account used for UDR deployment.	ocudr-serviceaccount	NA	NA
global.preInstall.image.name	Image name for the nudr-bulk-import-prehook pod which will take care of DB and table creation for UDR deployment.	nudr_bulk_import_pre_install_hook	NA	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.preInstall.image.tag	Image version for nudr-bulk-import-pre-install-hook pod image	25.1.201	NA	NA
global.preInstall.createUser	Creates user and grants access to UDR database.	false Note: Do not change this value.	true or false	NA
global.preInstall.config.logLevel	Log level for preinstall hook pod	WARN	Possible Values - WARN INFO DEBUG	NA
global.postUpgrade.image.name	Image name for the nudr-bulk-import-post-upgrade pod which will take care of DB and table creation for UDR deployment.	nudr_bulk_import_post_upgrade_hook	NA	NA
global.postUpgrade.image.tag	Image version for nudr_bulk_import_post_upgrade_hook pod image	25.1.201	NA	NA
global.postUpgrade.config.logLevel	Log level for postUpgrade hook pod	WARN	Possible Values - WARN INFO DEBUG	NA
global.hookJobResources.limits.cpu	CPU limit for pods created k8s hooks and jobs created as part of UDR installation. Applicable for helm test job too.	1	NA	NA
global.hookJobResources.limits.memory	Memory limit for pods created k8s hooks/jobs created as part of UDR installation. Applicable for helm test job as well.	1Gi	NA	NA
global.hookJobResources.limits.ephemeral-storage	Ephemeral Storage limits for hook pods	1024Mi	Unit MB	

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.hookJobResources.requests.cpu	CPU requests for pods created k8s hooks/jobs created as part of UDR installation. Applicable for helm test job as well.	0.5	NA	The cpu to be allocated for hooks during deployment
global.hookJobResources.requests.memory	Memory requests for pods created k8s hooks/jobs created as part of UDR installation. Applicable for helm test job as well	0.5Gi	NA	The memory to be allocated for hooks during deployment
global.hookJobResources.requests.ephemeral-storage	Ephemeral Storage request for hook pods	72Mi	Unit MB	NA
global.hooksCommonConfig.tolerations	When global.tolerationSetting is ENABLED configure tolerations here.	[]	Example: tolerations: - key: "key1" - operator: "Equal" - value: "value1" - effect: "NoSchedule"	NA
global.hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion	Node Selector API version setting	v1	Allowed Values: - v1 - v2	
global.hooksCommonConfig.nodeSelector.nodeKey	NodeSelector key configuration for UDR hooks, This excludes the common service hooks When hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration will come into use.	' '	Valid key for a label for a particular node	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
global.hooksCommonConfig.nodeSelector.nodeValue	NodeSelector Value configuration for UDR hooks. This excludes the common service hooks. When hooksCommonConfig.hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion is set to v1 then this configuration will come into use.	' '	Valid value pair for the above key for a label for a particular node	NA
global.hooksCommonConfig.nodeSelector	NodeSelector configuration when hooksCommonConfig.helmBasedConfigurationNodeSelectorApiVersion at micro service 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
dbCredSecretName	Database credential secret name	ocudr-secrets	NA	NA
customExtension.allResources.labels	Custom Labels that needs to be added to all the UDR Kubernetes resources	null	NA	NA
customExtension.allResources.annotations	Custom annotations that needs to be added to all the UDR Kubernetes resources	null	NA	ASM related annotations to be added under ASM Specific Configuration section
customExtension.lbServices.labels	Custom labels that needs to be added to UDR services that are considered as Load Balancer type	null	NA	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
customExtension.lbServices.annotations	Custom Annotations that needs to be added to OCUDR Services that are considered as Load Balancer type	null	NA	NA
customExtension.lbDeployments.labels	Custom labels that needs to be added to UDR deployments that are associated to a service of Load Balancer type	null	NA	NA
customExtension.lbDeployments.annotations	Custom annotations that needs to be added to UDR deployments that are associated to a service of Load Balancer type	null	NA	ASM related annotations to be added under ASM Specific Configuration section
customExtension.nonlbServices.labels	Custom labels that needs to be added to UDR services that are not considered as Load Balancer type	null	NA	NA
customExtension.nonlbServices.annotations	Custom annotations that needs to be added to UDR services are not considered as Load Balancer type	null	NA	NA
customExtension.nonlbDeployments.labels	Custom labels that needs to be added to UDR deployments that are associated to a service which is not of Load Balancer type	null	NA	NA
customExtension.nonlbDeployments.annotations	Custom annotations that needs to be added to UDR deployments 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

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
extraContainerTpl	The template for extraContainer is defined in this configuration. The debugContainer template defined here which is applicable for all microservices under ocudr umbrella.	<pre> - command: - /bin/ sleep - infinity image: {{ .Values.global.dockerRegistry }}/ debug- tools:25.1.20 3 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" cpu: "0.5" memory: "4Gi" </pre>	NA	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
		securityContext: allowPrivilegeEscalation: true capabilities: drop: - ALL add: - NET_RAW - NET_ADMIN runAsUser: 1002 volumeMounts: - mountPath: /tmp/tools name: debug-tools-dir		
extraContainersVolumesTpl	Debug container volume template	- name: debug-tools-dir emptyDir: medium: Memory sizeLimit: "4Gi"	NA	NA
image.name	Docker Image name	nudr_bulk_import	NA	NA
image.tag	Tag of Image	25.1.201	NA	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
image.pullPolicy	This setting indicates whether image needs to be pulled or not	IfNotPresent	Possible Values - Always IfNotPresent Never	NA
createBulkImportPVC.name	name of PVC	bulkimportpersistentclaim	NA	NA
createBulkImportPVC.storageClassName	storage class name for PVC	standard	Possible Values - standard	NA
createBulkImportPVC.storage	storage for PVC	10Gi	NA	NA
createXmlToCsvPVC.name	name of PVC	xmltocsvpersistentclaim	NA	NA
createXmlToCsvPVC.storageClassName	storage class name for PVC	standard	Possible Values - standard	NA
createXmlToCsvPVC.storage	storage for PVC	10Gi	NA	NA
xmlToCsv.enabled	flag for enabling or disabling nudr-xmltocsv	false	true or false	NA
xmlToCsv.management	management port of xmltocsv container	9001	NA	NA
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 notify service to receive signals from nudr-notify-service pod.	5001	NA	NA
service.port.management	The actuator management port to be used for notify service.	9000	NA	NA
hikari.poolsize	The Hikari pool connection size to be created at start up	10	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

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
hikari.idleTimeout	Idle MYSQL connections timeout	10000	Unit: Milliseconds	NA
hikari.queryTimeout	Query timeout for a single database query. If you set the value to -1 there will be no timeout for database queries	-1	Unit: Milliseconds	NA
logging.level.root	Log level of the notify service pod	WARN	Possible Values - WARN INFO DEBUG	NA
resources.requests.cpu	The CPU to be allocated for notify service pod during deployment	2	NA	NA
resources.requests.memory	The memory to be allocated for nudr-bulk-import pod during deployment	2Gi	NA	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	NA
resources.limits.memory	Memory allotment limitation	2Gi	NA	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 auto scaling (creating HPA)	80	NA	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

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
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	false	NA	Do not change this value unless it is different from the value configured in global section.
istioSidecarReady Url	Readiness URL configurable for side car, referred to global section istioSidecarReady Url. Change only if the url is different for the side car container in this microservice	http://127.0.0.1:15000/ready	NA	Do not change this value unless it is different from the value configured in global section.
ocudrReleaseName	Release Name of UDR to form config-server/ocudr endpoint FQDN	ocudr	NA	NA
pdbiToCsv.enabled	If this parameter is enabled, then only pdbiToCsv container is used to convert PDBI format to CSV format. Otherwise, not.	false	Possible Values - true or false	NA
pdbiToCsv.managementPort	Management Port of PDBI to CSV container	9002	NA	NA
createPdbiToCsvPVC.name	Name of Persistent Volume Claim (PVC) for PDBI to CSV	pdbitocsvpersistentclaim	NA	NA
createPdbiToCsvPVC.storageClassName	Storage class name for PVC	standard	Possible Values - standard	NA
createPdbiToCsvPVC.storage	Storage for PVC	10Gi	NA	NA
createPdbiToCsvPVC.pathLocation	Directory from where PDBI files are polled	/home/udruser/pdbi	NA	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
sftpDetails.enable	To enable the sftp feature on bulk-import	false	Possible Values - true or false	NA
sftpDetails.host	Remote Host IP	10.75.229.18	NA	NA
sftpDetails.port	Remote Host Port	22	NA	NA
sftpDetails.userName	Remote Host Username	cloud-user	NA	NA
sftpDetails.path	Remote Host server path where .pdbi files present	home/cloud-user/ocudr/sftppath/	NA	Do not delete the backslash / in the default value. It is mandatory to include the backslash. This is also applicable during the REST API operation.
sftpDetails.transferOutPath	Remote Host server path where generated .log files are placed	/home/cloud-user/transferOut/	NA	NA
sftpDetails.extension	Extension of the files to be transferred, as of now support will be for .pdbi	.pdbi	.pdbi	NA
sftpDetails.pollingImport	Periodic lookup interval for sftp transfer-in	"2"	NA	NA
sftpDetails.pollingExport	Periodic lookup interval for sftp transfer-out	"1"	NA	NA
sftpDetails.deleteLogFilePvc	This flag is used to delete the log files from pvc path once the file transfer is successfully done to the remote host	false	Possible Values - true or false	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

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
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
maxConcurrentPushedStreams	Maximum number of concurrent requests that is pushed per destination	6000	NA	NA
maxRequestsQueuedPerDestination	Maximum number of requests that is 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	NA	NA
idleTimeout	Client connection idle timeout	0	NA	NA
dnsRefreshDelay	Server DNS refresh delay	120000	NA	NA
pingDelay	Server ping delay	60	NA	NA
connectionFailureThreshold	Client connection failure threshold	10	NA	NA
jettyServiceMeshCheck	For ASM deployment jetty service mesh check	false	true/false	NA
batchConfig.asynchronous	To enable multiple thread bulk import and single thread bulk import	false	true/false	NA
batchConfig.coreThreadPool	Number of minimum and core thread pool size for spring batch	2	NA	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
batchConfig.maxThreadPool	Number of maximum thread pool size for spring batch	2	NA	NA
batchConfig.queueCapacity	Number maximum queue size	2000	NA	NA
batchConfig.writerCoreThreadPool	Number of minimum or core thread pool size for task executor	5	NA	NA
batchConfig.writerMaxThreadPool	Number of maximum thread pool size for task executor	5	NA	NA
batchConfig.writerQueueCapacity	Maximum queue size of task executor	2000	NA	NA
xmlToCsv.batchConfig.coreThreadPool	Number of minimum and core thread pool size for spring batch	10	NA	NA
xmlToCsv.batchConfig.maxThreadPool	Number of maximum thread pool size for spring batch	10	NA	NA
xmlToCsv.batchConfig.queueCapacity	Maximum queue size	2000	NA	NA
xmlToCsv.batchConfig.chunkSize	Reading file chunk size	500	NA	NA
pdbiToCsv.batchConfig.coreThreadPool	Number of minimum and core thread pool size for spring batch	5	NA	NA
pdbiToCsv.batchConfig.maxThreadPool	Number of max thread pool size for spring batch	5	NA	NA
pdbiToCsv.batchConfig.queueCapacity	Maximum queue size	2000	NA	NA
pdbiToCsv.batchConfig.chunkSize	Reading file chunk size	500	NA	NA
pdbiToCsv.mandatoryFields.create	Mandatory field for create operation import though PDBI	imei	Valid field	NA
pdbiToCsv.mandatoryFields.update	Mandatory field for update operation import though PDBI	imei	Valid field	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
pdbiToCsv.mandatoryFields.delete	Mandatory field for delete operation import though PDBI	imei	Valid field	NA
pdbiToCsv.allFields.create	Fields list for create operation import though PDBI	iid,imei,svn,allow,white,greys,gray,black,block,imsi,msisdn,force,timeout	Valid field	NA
pdbiToCsv.allFields.update	Fields list for update operation import though PDBI	iid,imei,svn,allow,white,greys,gray,black,block,imsi,msisdn,force,timeout	Valid field	NA
pdbiToCsv.allFields.delete	Fields list for delete operation import though PDBI	iid,imei,imsi,msisdn,timeout,svn	Valid field	NA
sftpDetails.fileSizeCheck	Check the file size of the remote server once the file is successfully transferred to the remote location. This flag is used check file size of transfer out log files from PVC to remote location.	false	true or false	NA
sftpDetails.transferProtocol	Transfer the files to remote location. The default protocol is SFTP.	sftp	sftp or scp	NA
sftpDetails.secrets.privatekey.name	SFTP transfer private key holding secret.	ocudr-ssh-private-key	Valid K8s secret name	NA
sftpDetails.secrets.publickey.name	SFTP transfer public key holding secret.	ocudr-ssh-public-key	Valid K8s secret name	NA
gracefulShutdown.gracePeriod	It is the graceful shutdown time limit.	30s	Unit: Seconds	NA
deployment.replicaCount	Configures the number of pod replicas to deploy.	1	Positive integers	The deployment is StatefulSet.
deployment.customExtension.labels	Custom labels that must be added to specific subscriber bulk import deployment resource.	{ } (empty flow mapping)	Valid yaml mapping	NA

Table 2-51 (Cont.) Subscriber Bulk Import Tool User Parameters

Parameter	Description	Default value	Range or Possible Values (If applicable)	Notes
deployment.customExtension.annotations	Custom annotations that must be added to specific subscriber bulk import deployment resource.	{ } (empty flow mapping)	Valid yaml mapping	NA

2.37.2.2 Initiating the Subscriber Bulk Import Tool

By default, the Subscriber Bulk Import tool polls the **/home/udruser/import** path every 5000 ms. Before initiating the tool, ensure the **nudr-bulk-import** pod is up and running.

To initiate the bulk import tool:

1. Run the following command to place the file inside the container:
`kubectl cp <csv file> <namespace>/<nudr-bulk-import pod name>:/home/udruser/import`
Example: `kubectl cp create_pcf.csv bulktest/bulktest-nudr-bulk-import-pgh2p:/home/udruser/import`
2. To verify whether the file is present inside the container or not, log in to the bash of container and run the following command:
`kubectl exec -it <nudrbulk-import pod name>-n <namespace> -- bash`
Example: `kubectl exec -it bulktest-nudr-bulk-import-pgh2p -n bulktest -- bash`

Note

After log in to bash, you can find the file under **/home/udruser/import** location.

```
bash-4.2$ pwd
/home/udruser/import
bash-4.2$ ls -l
-rw-r--r--. 1 root udruser 28471 Aug 19 09:33 create_pcf.csv
```

The Bulk Import tool processes each CSV file at a time and provisions the subscribers on 5G UDR. The summary report is available in the **/home/udruser/import** path with filename (<CSV_filename>.log).

```
bash-4.2$ pwd
/home/udruser/import
bash-4.2$ ls -l
-rw-r--r--. 1 root udruser 28471 Aug 19 09:33 create_pcf.csv
create_pcf.log
```

A sample summary report is as follows:

```
Import Tool Summary Report
Time taken for processing file 0m26s
Number Of Failed Record 0
Total Number of records fetched from csv files 4
Total Number success for create operation 4
Total Number of failures for create operation 0
Total Number success for modify operation 0
Total Number of failures for modify operation 0
Total Number success for delete operation 0
Total Number of failures for delete operation 0
Total Number of failures for service time out 0
List of failed keys
All Records Success
```

Note

The Bulk Import container logs contains the summary report, when the INFO logging level is enabled.

- Run the following command to copy the report from pod container:
`kubectl cp <pod name>:home/udruser/import/modify_slf.log <file name> -n <namespace>`

Example: `kubectl cp ocudr-nudr-bulk-import-8g8c8:home/udruser/import/modify_slf.log modify_slf.log -n myudr`

Note

- The metrics of bulk import tool can be observed on the Grafana dashboard. For more information about Bulk import metrics, see [nudr-bulk-import-service Metrics](#).
- To monitor the status of bulk import tool, see [Subscriber Bulk Import Status](#).

2.37.2.2.1 Using xmltocsv Converter

By default, the xmltocsv converter polls the **/home/udruser/xml** path every 5000 ms. Before using xmltocsv converter, ensure the nudr-bulk-import microservice is running with the nudr-bulk-import and nudr-xmltocsv containers. To use the xmltocsv converter:

- Run the following command to place the ixml file in the path (/home/udruser/xml) inside the container:

```
kubectl cp <csv file> <namespace>/<nudr-bulk-import pod name>:/home/udruser/xml -c nudr-xmltocsv
```

Example: `kubectl cp createSubscriber.ixml bulktest/bulktest-nudr-bulk-import-pgh2p:/home/udruser/xml -c nudr-xmltocsv`

- Log in to bash of the container and run the following commands to check the ixml file inside the container:
`kubectl exec -it <nudrbulk-import pod name> -c nudr-xmltocsv -n <namespace>`
`bash`

```
Example: kubectl exec -it bulktest-nudr-bulk-import-pgh2p -c nudr-xmltocsv -n
bulktest bash
```

```
bash-4.2$ pwd
/home/udruser/xml
bash-4.2$ ls -l
-rw-r--r--. 1 root udruser 28471 Aug 19 09:33 createSubscriber.xml
createSubscriber.log
```

Note

After log in to bash, you can find the file under **/home/udruser/xml** location.

The xmltocsv convertor processes the ixml file, converts XML format to JSON format, and then generate CSV file as output. This CSV file is automatically placed in the /home/udruser/import folder.

The bulk import container processes this CSV file and provisions the subscribers on 5G UDR. You can check this CSV file in the /home/udruser/import folder.

```
bash-4.2$ pwd
/home/udruser/import
bash-4.2$ ls -l
-rw-r--r--. 1 root udruser 28471 Aug 19 09:33 createSubscriber.csv
```

You can see the progress of XML file processing in logs only if the INFO logs are enabled. Run the following command to generate a summary report of xmltocsv convertor in the same path (/home/udruser/xml) with file name as **<CSV_filename>.log**.

```
cat createSubscriber.log
```

A summary report of xmltocsv container is created in same path(/home/udruser/xml) with filename **<CSV_filename>.log**.

```
bash-4.2$ pwd
/home/udruser/xml
bash-4.2$ ls -l
-rw-r--r--. 1 root udruser 28471 Aug 19 09:33 createSubscriber.xml
createSubscriber.log
```

A sample summary report which is created in xmltocsv container is as follows:

```
XML to CSV Tool Summary Report :
Time taken for processing file : 0m0s
Number Of Failed Record :0
Total Number of records fetched from xml file :100
Total Number success for create operation :100
Total Number of failures for create operation :0
Total Number success for modify operation :0
Total Number of failures for modify operation :0
Total Number success for delete operation :100
Total Number of failures for delete operation :0
```

List of failed keys :
All Records Success

A summary report of bulk import container is created in same path (/home/udruser/import) with file name as **<CSV_filename>.log**

A summary report of bulk import container is created in same path(/home/udruser/import) with filename **<CSV_filename>.log**

```
bash-4.2$ pwd
/home/udruser/import
bash-4.2$ ls -l
-rw-r--r--. 1 root udruser 28471 Aug 19 09:33 createSubscriber.csv
createSubscriber.log
```

A sample summary report created in bulk import is as follows:

```
Import Tool Summary Report
Time taken for processing file 0m26s
Number Of Failed Record 0
Total Number of records fetched from csv files 4
Total Number success for create operation 4
Total Number of failures for create operation 0
Total Number success for modify operation 0
Total Number of failures for modify operation 0
Total Number success for delete operation 0
Total Number of failures for delete operation 0
Total Number of failures for service time out 0
List of failed keys
All Records Success
```

This summary report is also available in the xmltocsv and bulk import containers logs when INFO logging level is enabled. To copy the report from a pod container to the current location (<file name>), run the following command:

```
kubectl cp <pod name>:home/udruser/xml/createSubscriber.log <file name> -c
nuds-xmiltocsv -n <namespace>
```

Example: `kubectl cp ocuds-nuds-bulk-import-8g8c8:home/udruser/xml/createSubscriber.log createSubscriber.log -c nuds-xmiltocsv -n myuds`

① Note

- The metrics of bulk import tool can be observed on the Grafana dashboard. For more information about bulk import metrics, see [nuds-bulk-import-service Metrics](#).
- You can monitor the xmltocsv metrics on Prometheus or Grafana.

2.37.2.3 Upgrade and Rollback support for Subscriber Bulk Import Tool

Upgrade Procedure

Helm upgrade is supported. To change the configuration for an existing Subscriber Bulk Import Tool, see [Change the Configuration for an Existing Subscriber Bulk Import Tool using Helm](#).

Run the below command to perform the subscriber bulk import tool upgrade.

```
helm upgrade <release name> -n <name space> <chart path>
```

Upgrade Prerequisites

- UDR must be upgraded first and then Subscriber Bulk Import Tool must be upgraded. For more information, see "Upgrading UDR" section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- Only existing container is enabled when upgrading. For example, if in the previous deployment only the nudr-bulk-import container is enabled then when upgrading you can only enable nudr-bulk-import container.
- PVC size cannot be modified as part of the upgrade process.

Rollback Procedure

Rollback instruction for Subscriber Bulk Import:

Note

Helm rollback is not supported for Subscriber Bulk Import Tool.

Perform the following steps to rollback Subscriber Bulk Import Tool:

Note

UDR and the Subscriber Bulk Import Tool must be rolled back together.

- Uninstall the current version of the Subscriber Bulk Import Tool.
- Install the older version of the Subscriber Bulk Import Tool. For more information, see [Installing Subscriber Bulk Import Tool](#).
- Perform the rollback procedure for UDR. For more information, see Rolling Back UDR chapter in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

2.37.2.4 Sample Subscriber Bulk Import File

This section provides sample subscriber bulk import files.

2.37.2.4.1 Sample Bulk Import File in CSV Format

The Bulk Import file should be in CSV file format, which is shared below:

Sample: CSV File Format

```
keyType,keyValue,operationType,nfType,jsonPayload
msisdn,1111111114,CREATE,PCF,{"profile-data": {"MSISDN":
["9111111112", "9211111112", "1111111114"]}, "policy-data": {"am-data":
{"praInfos": {"p1": {"praId": "p1", "trackingAreaList": [{"plmnId": {"mcc":
"976", "mnc": "32"}, "tac": "5CB6"}}, {"plmnId": {"mcc": "977", "mnc":
```

```
"33"},"tac": "5CB7"}], "ecgiList": [{"plmnId": {"mcc": "976", "mnc":
"32"},"eutraCellId": "92FFdBE"}, {"plmnId": {"mcc": "977", "mnc":
"33"},"eutraCellId": "8F868C4"}], "ncgiList": [{"plmnId": {"mcc": "976", "mnc":
"32"},"nrCellId": "b2fB6fE9D"}, {"plmnId": {"mcc": "977", "mnc":
"33"},"nrCellId": "5d1B4127b"}], "globalRanNodeIdList": [{"plmnId": {"mcc":
"965", "mnc": "235"}, "n3IwfId": "fFf0f2AFbFa16CEfE7"}, {"plmnId": {"mcc":
"967", "mnc": "238"}, "gNbId": {"bitLength": 25, "gNBValue":
"1A8F1D"}}}], "subscCats": ["cat1", "cat2"]}]}"
```

```
accountID,12345678912345678912345678,MODIFY,SLF,{"profile-data": {"imsi":
["2222222221", "2222222222"], "nai": ["test@abc.com"], "accountID":
["12345678912345678912345678"], "msisdn":
["19195225555", "19195225556"], "extid": ["user@abc.com"]}, "slfGroupName":
"IMSGrp1"}"
```

```
msisdn,63612000056,DELETE,PCF-VSA
```

```
msisdn,1111111114,DELETE,PCF
```

The bulk import tool supports following keyTypes:

Table 2-52 CSV File Format Details

keyType	keyValue	OperationType	nfType	Subs_data(JSON)
MSISDN	1234	CREATE	PCF	{ .. }
IMSI	6786677	MODIFY	PCF	{ .. }
NAI	abcd@oracle.com	DELETE	PCF	{ .. }
EXTID	user@abcd.com	CREATE	PCF	{ .. }
NAI	test@abcd.com	CREATE	PCF	{ .. }
IMEI	355171053475182	CREATE	EIR	{ .. }

Note

The **subs_data** column must have JSON body enclosed in double quotes("").

2.37.2.4.2 Sample iXML format for Bulk Import

The sample iXML format for bulk import is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction>
  <txRequest id="1">
    <createSubscriber>
      <key>
        <MSISDN>6666666100</MSISDN>
      </key>
      <entity>
        <data>
          <name>Subscriber</name>
          <interface>XMLIMPORT</interface>
        </data>
        <content><![CDATA[<subscriber><field name="MSISDN">6666666100</
```

```

field><field name="IMSI">2222222222222222</field><field name="BillingDay">31</
field><field name="Entitlement">DayPass</field><field
name="Entitlement">DayPassValue</field></subscriber>]]></content>
    </entity>
  </createSubscriber>
</txRequest>
</transaction>

```

2.37.2.4.3 Sample Bulk Import File in CSV Format for EIR

The bulk import file should be in CSV file format, as below:

Sample: CSV File Format

```

keyType,keyValue,operationType,nfType,jsonPayload,addInfo
IMEI,355171053475182,MODIFY,EIR,"{"cmd":"ent_eir(iid 3237536, imei
355171053475182, black yes, force yes)","profile-data":{"imei":
["355171053475182"]},"equipment-status":
{"white":false,"gray":false,"black":false}}"

keyType,keyValue,operationType,nfType,jsonPayload,addInfo
IMEI,356508052681576,DELETE-KEY,EIR,"{"cmd":"dlt_eir(iid 3237532, imei
356508052681576, imsi 12345678)","profile-data":{"IMSI":["12345678"]}}"

```

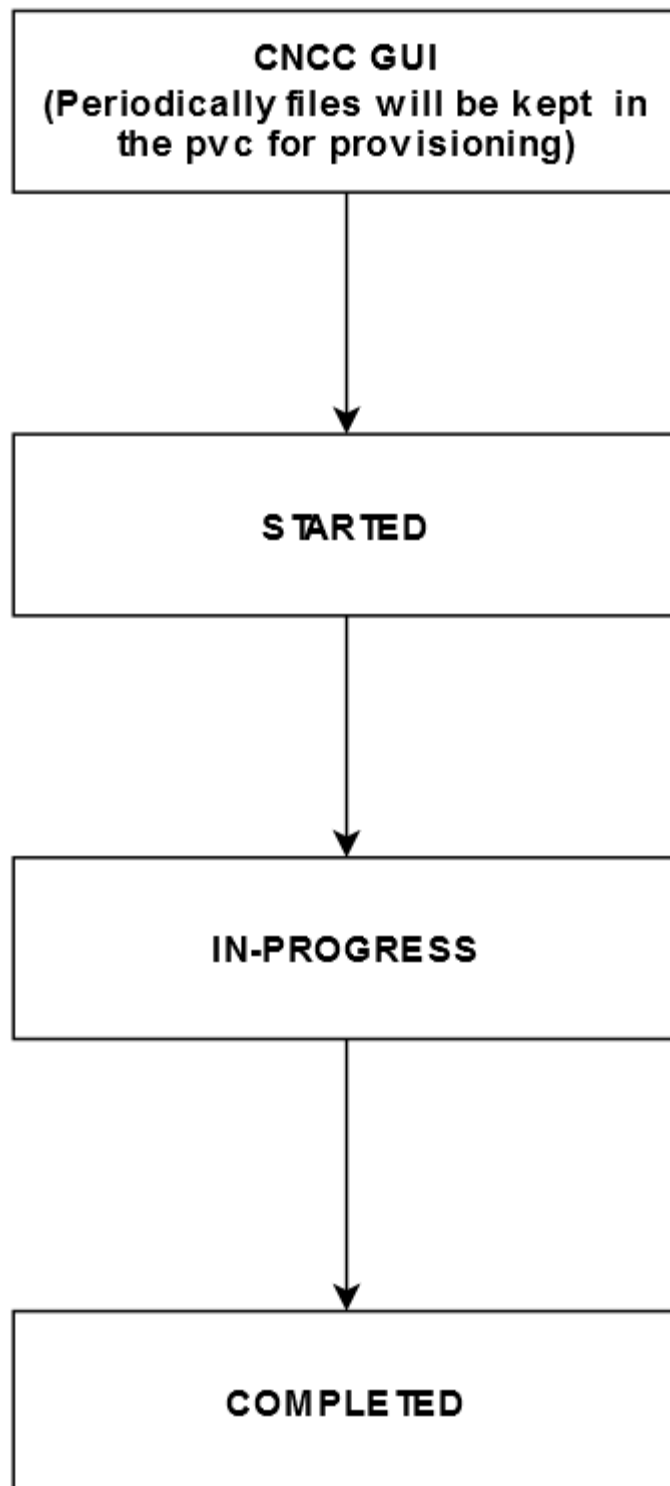
2.37.3 Monitoring the Subscriber Bulk Import Tool Status

Monitoring the Subscriber Bulk Import Tool Status for UDR and SLF Mode

The following are the subscriber bulk import tool status during operation:

- When the file is copied to the PVC path, the subscriber bulk import tool shows the status as **STARTED**.
- When the subscriber bulk import tool picks the file from the PVC, the status is shows as **IN-PROGRESS**.
- When the provisioning is complete, the status shows **COMPLETED** and the result log file is generated and placed in the PVC path.

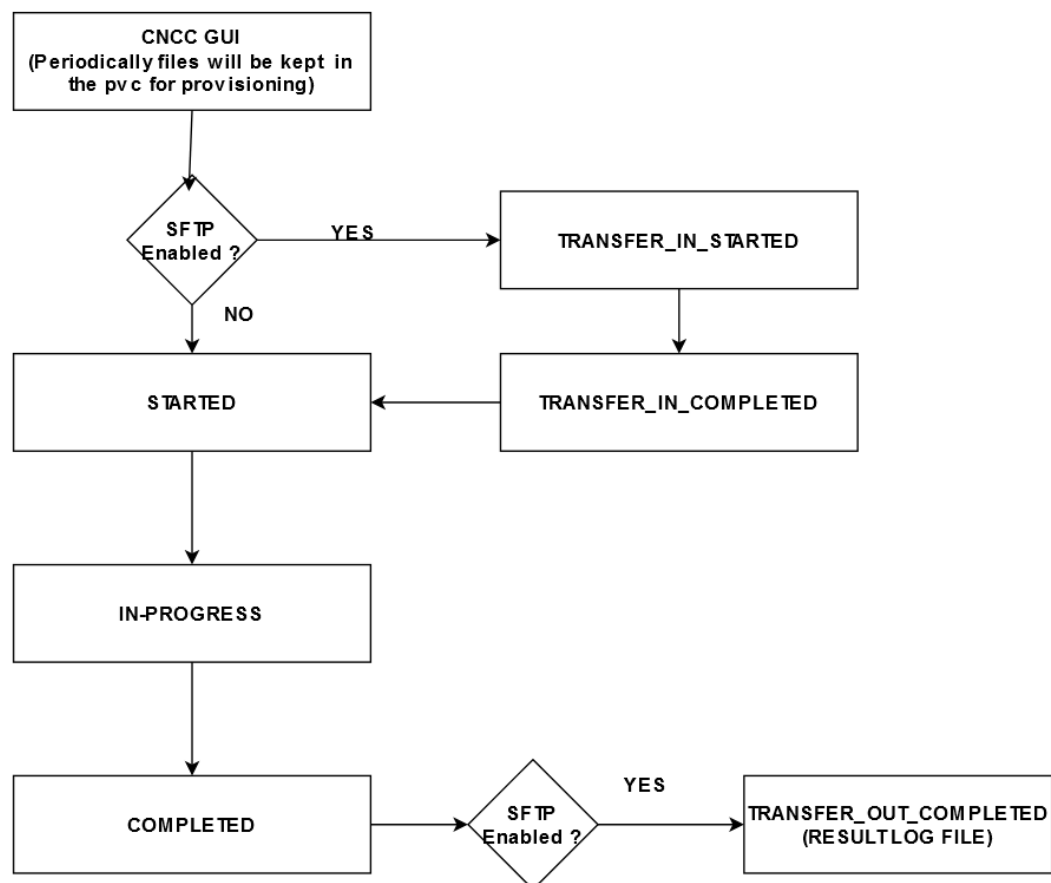
Figure 2-38 Bulk Import Tool Status for UDR and SLF

**Monitoring the Subscriber Bulk Import Tool Status for EIR Mode**

The following are the subscriber bulk import tool status during operation:

- If SFTP is enabled in the subscriber bulk import tool, the status shows **TRANSFER_IN_STARTED**. When the file transfer is complete, the status shows **TRANSFER_IN_COMPLETED** and provisioning starts.
- If SFTP disabled in the subscriber bulk import tool, the status is shown as **STARTED**.
- When the file is present in the PVC, the subscriber bulk import tool will pick the file and the status shows **IN-PROGRESS**.
- When the provisioning is completed, the status shows **COMPLETED** and the result log file is generated and placed in the PVC path.
- If SFTP is enabled in the subscriber bulk import tool, the result log file is transferred to the remote host and the status shows **TRANSFER_OUT_COMPLETED**.

Figure 2-39 Bulk Import Tool Status for EIR



For more information, see the CNC Console section [Subscriber Bulk Import Status](#).

2.37.4 Secure File Transfer Support for Subscriber Bulk Import Tool

Subscriber bulk import tool supports the capability to securely transfer the files from the remote server using Secure File Transfer Protocol (SFTP). This feature supports transferring-in the Provisioning Database Interface (PDBI) import files and transferring out the result log files to a remote server.

Using Helm Configuration:

You can enable or disable secure file transfer support using Helm Configuration.

- You can enable or disable the SFTP by setting the `sftpDetails.enable` parameter to true or false in `custom-value.yaml` file of the subscriber bulk import tool. For more information on the parameters, see [Configuring Subscriber Bulk Import Tool](#).

```
sftpDetails:
  enable: true
  host: 10.75.229.127
  port: 22
  username: cloud-user
  path: /home/cloud-user/ocudr/sftppath1/
  extension: .pdbi
  pollingImport: "2"
  pollingExport: "1"
  deleteLogFilePvc: false
  secrets:
    privatekey:
      name: ocudr-ssh-private-key
    publickey:
      name: ocudr-ssh-public-key
    fileSizeCheck: false
    transferProtocol: sftp
```

- Configure the private and public keys for the remote host. The operator must provide the private key (`id_rsa`) and the public key (`id_rsa.pub`) for the remote host.

```
kubectl create secret generic ocudr-ssh-private-key --from-file=id_rsa=/
home/cloud-user/ocudr/secrets/id_rsa -n <namespace>
kubectl create secret generic ocudr-ssh-public-key --from-file=id_rsa.pub=/
home/cloud-user/ocudr/secrets/id_rsa.pub -n <namespace>
```

When the SFTP is enabled the scheduler looks for the `.pdbi` files from the `remoteHost` for the path that is configured under the `remoteDirPath` parameter and transfers the files to the PVC location. When the files processing are completed the resultant log files are transferred to the same `remoteHost` from PVC location to a separate path that is configured under the `transferOutPath` parameter.

Using REST API:

You can perform the PUT operation on the `http://10.75.215.124:32523/nudr-config/v1/udr.bulkimportservice.cfg/BULK_IMPORT_CFG` to configure the SFTP feature. For more information, see [Configuring Subscriber Bulk Import Tool](#) and *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

```
{
  "pollingTimeInterval": 5000,
  "lastModifiedTimeInterval": 30,
  "pdbiFilePath": "/home/udruser/pdbi",
  "pdbiFileExpiryTime": "7d",
  "remoteHost": "10.75.229.128",
  "remoteHostPort": "22",
  "remoteDirPath": "/home/cloud-user/ocudr/sftppath/",
```

```

    "remoteHostUsername": "cloud-user",
    "fileExt": ".pdbi",
    "sftpEnabled": true,
    "deleteFileFromPVC": false,
    "periodicImportPoll": "2",
    "periodicExportPoll": "1",
    "resultLogFilePath": "/home/udruser/pdbi/logs"
    "transferOutPath": "/home/cloud-user/transferOut/"
  }

```

When the keys are generated by the operator for the remote host, you must run the below command to create a kubernetes secrets for both keys.

```

kubectl create secret generic ocudr-ssh-private-key --from-file=id_rsa=/home/
cloud-user/ocudr/secrets/id_rsa -n <namespace>
kubectl create secret generic ocudr-ssh-public-key --from-file=id_rsa.pub=/
home/cloud-user/ocudr/secrets/id_rsa.pub -n <namespace>

```

2.38 Alternate Route Service

Overview

UDR uses Alternate Route Service to connect to other NFs, especially to NRF, for egress traffic. Alternate Route service supports static and dynamic configurations of virtual FQDN of NRF.

By using Alternate Route Service, you can configure N number of destination url's as per your choice. **Example:** There are two NRFs, NRF1 and NRF2. If NRF-1 goes down and it redirect the requests to the next NRF, 'NRF2' based on priority and weights and so on until a working NRF is found.

Note

Before routing to NRF2, UDR retries NRF1 configured number of times and then, route to NRF2.

To Enable

To enable Alternate Route Service, set the **alternateRouteServiceEnable** flag as 'true' under the Alternate Route section of values.yaml file.

To Configure

You can configure Alternate Route Service in the **nrf-client-service** section of the Values.yaml file as shown below:

Figure 2-40 Configuring Alternate Route Service

```

alternateRouteVirtualFqdn: abc.test.com
alternateRouteSchema: http
dnsSrvEnabled: true
dnsSrvMaxRetryCount: 1
alternateRoutePort: 80

```

This service runs in a separate microservice in which you need to configure virtual FQDN as shown below:

Figure 2-41 Configuring Virtual FQDN

```
#Static virtual FQDN Config
staticVirtualFqdns:
  - name: http://abc.test.com
    alternateFqdns:
      - target: abc.test.com
        port: 8080
        priority: 10

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

#Below configuration is for customizing the format of FQDN which will used while querying coreDNS for SRV Records
dnsSrvFqdnSetting:
  enabled: true #If this flag is disabled, then default value of "{scheme}._tcp.{fqdn}." will be used for Pattern
  pattern: "{scheme}._tcp.{fqdn}." #Ex: _http._tcp.service.example.org.
```

2.39 3gpp-Sbi-Correlation-Info Header

The implementation of **3gpp-Sbi-Correlation-Info** header enables network analytic tools or probes to easily identify the messages exchanged for a given User Equipment (UE). The Network Function (NF) service consumer or NF service producer can include any one of the UE identifiers (such as imsi, msisdn, extiid) in its 3gpp-Sbi-Correlation-Info header. The identifier helps to identify the UE related to the HTTP request or response.

If the HTTP request contains the 3gpp-Sbi-Correlation-Info header, then the HTTP server (if it supports this header) includes the header details in the response and sends it to the HTTP client with the same UE identity that was present in the 3gpp-Sbi-Correlation-Info header of the received HTTP request.

If the HTTP request creates a notification subscription and the HTTP server supports this header, then the HTTP server:

- Stores the UE identifier received in the header
- Includes the header in subsequent callback notification requests

To retrieve or insert information, the **3gpp-Sbi-Correlation-Info** header ensures that:

- There exists only one entry for its header.
- The header has only one identifier from the following list:
 - imsi
 - msisdn
 - extiid
 - accountid
 - nai
 - impi
 - suci
 - gci
 - gli
 - impu
 - imei

- imeisv
- mac
- eui

Note

Unified Data Repository (UDR) supports only imsi, msisdn, extid, accountid, and nai identifiers.

To Enable

To enable the **3gpp-Sbi-Correlation-Info** header, set the `sbiCorrelationInfoEnable: true`. You can use any one of the following ways to enable the **3gpp-Sbi-Correlation-Info** header.

- REST API: Set the **enabled** value to 'true' in request JSON of the Options configuration API. For more information, see the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.
- User Configurable Parameter: Set the **sbiCorrelationInfoEnable** parameter to 'true'. For more information, see the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- CNC Console: In the **Global Configurations** page, set the **SBI Correlation Info Enable** field to 'true'. For more information, see the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

For all success responses (2xx), the **3gpp-Sbi-Correlation-Info** header is added by default. For all error responses (4xx and 5xx), the **3gpp-Sbi-Correlation-Info** header is added only if the request has a header.

To Maintain

In case of failure or misconfiguration, the **3gpp-Sbi-Correlation-Info** header does not populate in UDR or Subscriber Location Function (SLF) responses. It is recommended to follow the instructions shared in the [To Enable](#) section.

2.39.1 3gpp-Sbi-Correlation-Info Header for Ingress Gateway

For the incoming requests, the nudr-dr service of UDR fetches the correlation header value from R-URI (Request - Uniform Resource Identifier) of the nudr-dr service and populates it into the correlation header of the nudr-dr service response. UDR ignores the correlation header value in the request and uses the R-URI value for the correlation header value in the response.

If the correlation header is not present in the incoming requests, then it is not sent in the error responses and the correlation header is populated with the identity used for subscriber data retrieval from the UDR database.

Example: **Header: 3gpp-sbi-correlation-info: imsi-4411111114**

Note

If the nudr-dr microservice is slow or down, then Ingress Gateway rejects the requests and does not add the correlation header.

The correlation header is added for all the success and failures responses of UDR, SLF, or EIR for only signaling APIs such as "nudr-dr" and "nudr-group-id-map". The list of APIs that support 3gpp-Sbi-Correlation-Info header are as follows:

Table 2-53 APIs Supporting 3gpp-Sbi-Correlation-Info Header

Resource	URI
policy-data	{apiRoot}/nudr-dr/{apiVersion}/policy-data/ues/{ueld}/am-data
-	{apiRoot}/nudr-dr/{apiVersion}/policy-data/ues/{ueld}/sm-data
-	{apiRoot}/nudr-dr/{apiVersion}/policy-data/ues/{ueld}/ue-policy-set
-	{apiRoot}/nudr-dr/{apiVersion}/policy-data/ues/{ueld}/sm-data/{usageMonId}
subscription-data	{apiRoot}/nudr-dr/{apiVersion}/subscription-data/{ueld}/authentication-data/authentication-subscription
-	{apiRoot}/nudr-dr/{apiVersion}/subscription-data/{ueld}/authentication-data/authentication-status
-	{apiRoot}/nudr-dr/{apiVersion}/subscription-data/{ueld}/context-data
-	{apiRoot}/nudr-dr/{apiVersion}/subscription-data/{ueld}/context-data/amf-3gpp-access
-	{apiRoot}/nudr-dr/{apiVersion}/subscription-data/{ueld}/context-data/amf-non-3gpp-access
-	{apiRoot}/nudr-dr/{apiVersion}/subscription-data/{ueld}/context-data/smf-registrations
-	{apiRoot}/nudr-dr/{apiVersion}/subscription-data/{ueld}/context-data/smf-non-3gpp-access
nf-group-ids	{apiRoot}/nudr-group-id-map/{apiVersion}/nf-group-ids?subscriber-id=msisdn-1111111111&nf-type=UDM,AUSF,NEF

2.39.2 3gpp-Sbi-Correlation-Info Header for Egress Gateway

If the incoming request creates a subscription to notifications (subs-to-notify) and has 3gpp-Sbi-Correlation-Info header, then UDR stores this header in its database and includes the stored header in subsequent callback notification requests.

For all the provisioning or update resource requests and Network Function (NF) resource update requests subscribed for notifications, the 3gpp-Sbi-Correlation-Info header is retrieved from the database and added in the PCF or UDM notification requests if subscription to notifications (subs-to-notify) has 3gpp-Sbi-Correlation-Info header. All the responses from Egress Gateway are validated irrespective of presence of header and respective metric is being pegged.

The APIs that support 3gpp-Sbi-Correlation-Info header are as follows:

Table 2-54 APIs Supporting 3gpp-Sbi-Correlation-Info Header

Resource	URI
subs-to-notify	https://pcf12.example.com/serviceY/abc

For more information about the APIs that support this feature, see the **Subscription Notification Formats for PCF Data** section in the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

2.40 Automated Test Suite Support

UDR provides Automated Test Suite (ATS) for validating different functionalities. ATS allows you to execute UDR, SLF, and EIR test cases using an automated testing tool, and then compares the actual results with the expected or predicted results. In this process, there is no intervention from the user is required. For more information, see *Oracle Communications Cloud Native Core, ATS User Guide*.

3

Configuring Unified Data Repository using CNC Console

This section describes how to configure global and service parameters on UDR using CNC Console. The service configuration includes data repository service, notify service, NRF client service and diameter service. Through CNC Console, you can view the bulk import status and logging level details, perform provisioning operations for profile data, PCF data, SLF data, UDM data and schema management and validate the access token.

3.1 Logging into CNC Console

After logging successfully into the CNC Console application, you can use UDR integrated with CNC Console. To log in successfully into the CNC Console, edit the hosts file available at the **C:\Windows\System32\drivers\etc** location.

In the Windows system, open the **hosts** file in the notepad as an Administrator and append the following set of lines at the end:

Example:

```
10.75.212.88 cncc-iam-ingress-gateway.cncc.svc.cluster.local  
10.75.212.88 cncc-core-ingress-gateway.cncc.svc.cluster.local
```

Note

The IP Address in the above lines may change when deployment cluster changes.

Save and close the notepad.

Note

Before logging to CNC Console, it is important to create a CNC user. Using this user details, you can log in to the CNC Console application. For information on creating a CNC Console user, you can refer to its user guide.

To log in to CNC Console:

1. Type **`http://cncc-core-ingress-gateway.cncc.svc.cluster.local:<PortNumber>/`** in the web browser and press Enter. The login page appears:

Figure 3-1 CNC Console Login Page

ORACLE®

Login to CNC Console

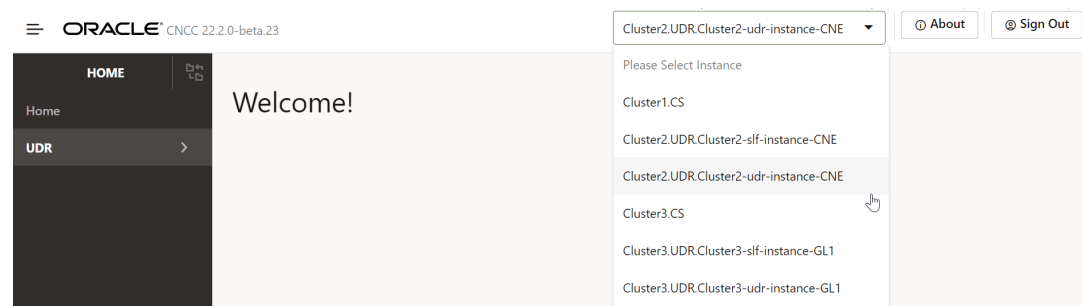
Username or email

Password

Log In

2. Enter the **Username** and **Password** and click **Log In**. The CNC Console Home page appears from where a user can select different instances in the drop down, for example, the EIR, SLF, or the UDR instance.

Figure 3-2 CNC Console Home Page

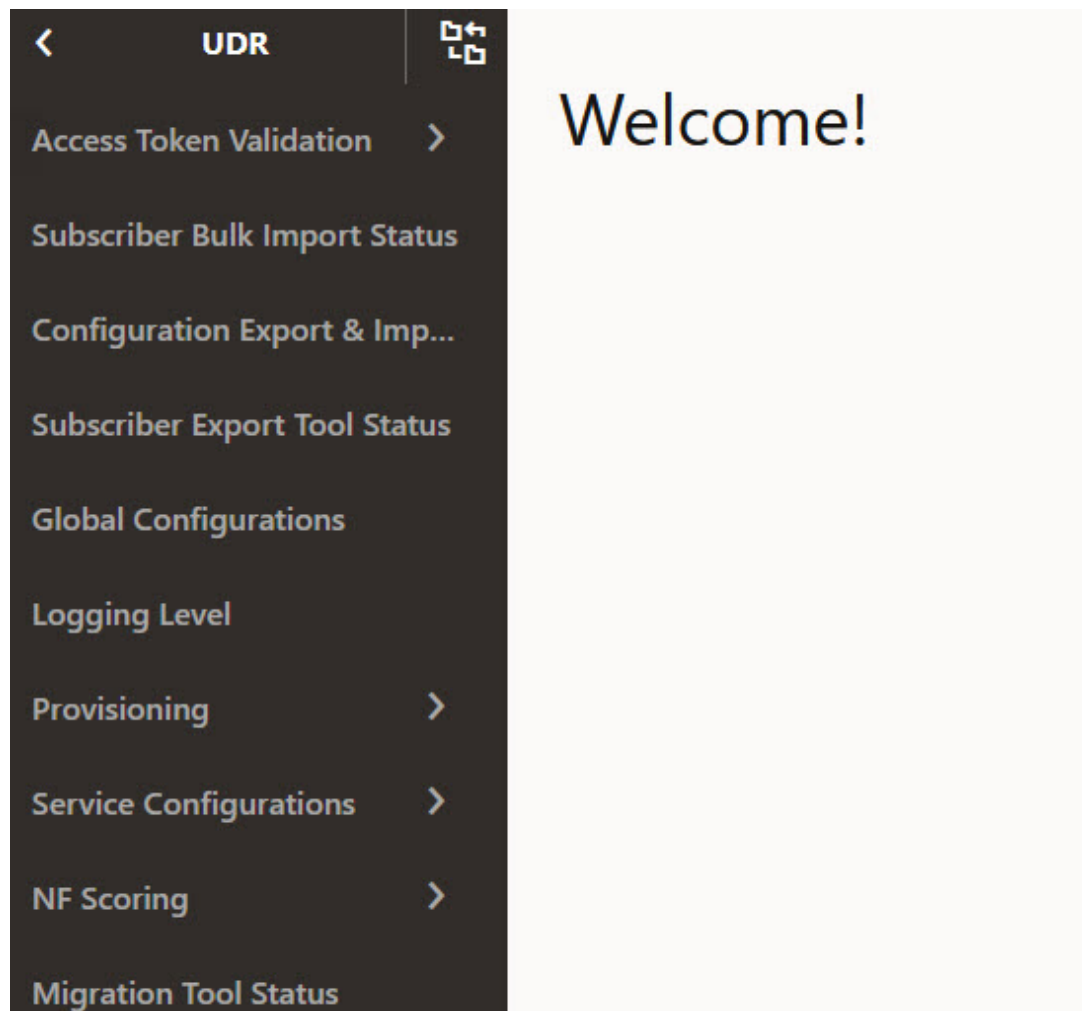


The following NF services appear in the left navigation pane for the selected instance:

- Access Token Validation
- Bulk Import Status
- Configuration Export & Import
- Global Configurations
- Logging Level

- Provisioning
 - Service Configurations
 - NF Scoring
 - Migration Tool Status
3. Select the service to be used from the left navigation pane:

Figure 3-3 CNC Console Home Page with Services



Note

The NF services appear in the left navigation pane for the EIR and SLF instance.

3.2 Egress Gateway

This section describes the Egress Gateway configurations.

1. In the left navigation pane, click **UDR** and then **Egress Gateway**.

2. Using **Egress Gateway**, you can configure following services:

- [Configurable Error Codes](#)
- [Error Code Profile](#)
- [Peer Configuration](#)
- [Peer Set Configuration](#)
- [Routes Configuration](#)
- [SBI Error Action Sets](#)
- [SBI Error Criteria Sets](#)
- [User Agent Header Generation](#)

3.2.1 Configurable Error Codes

Perform the following procedure to configure error codes using the CNC Console:

1. In the CNC Console, click **UDR** and then **Ingress Gateway** or **Egress Gateway**
2. For Ingress Gateway only, from the left navigation menu, navigate to the **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click the **Configurable Error Codes** option.
The **Configurable Error Codes** page is displayed with default values.
4. Click **Edit** to update the **Configurable Error Codes** configuration.
The **Edit Configurable Error Codes** page is displayed.
5. Click **Add** .
The **Add Error Scenarios** page is displayed.
6. Update the configuration details as required and click **Save**.
A confirmation message, "Save successfully." appears.
.
7. Click **Edit** .
The **Edit Error Scenarios** page is displayed.
8. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully." appears.
9. If you do not want to modify any configuration, click **Cancel**.
10. Click **Delete** and click **OK** to delete the selected configuration.
11. Click **Save** on the **Edit Configurable Error Codes** page to save the details. Click **Cancel** to discard your progress and go back to **Configurable Error Codes** page.
12. Click **Refresh** to reload the **Configurable Error Codes** page.

3.2.2 Error Code Profile

This configuration is used to update the errorcodeprofiles that is used in Overload Control, Controlled Shutdown feature for populating details in error responses when a request is discarded.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Ingress Gateway** or **Egress Gateway**
2. For Ingress Gateway only, from the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click **Error Code Profiles** option.
The **Error Code Profiles** page is displayed with default values for few fields.
4. Click **Add** to add the profiles.
The **Create Error Code Profiles** page is displayed.
5. Configure the fields in the **Create Error Code Profiles** page as mentioned in [Table 3-1](#).
6. Click **Save** on the **Create Error Code Profiles** page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Profiles** page.

Note

- Use the **Edit** icon available in the next column of the specific entry to update the route error profile information.
- Use **Refresh** icon to refresh the route error profile information.

Table 3-1 Error Code Profile Configuration

Field Name	Description	Details
Name	Indicate the error name.	DataType: string Constraints: NA Default Value: NA Presence: M
Error Code	This field determines the HttpStatusCode that needs to be populated in ProblemDetails (HttpStatus field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: M
Error Cause	This field determines the error cause that needs to be populated in ProblemDetails (Cause field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Title	This field determines the title that needs to be populated in ProblemDetails (Title field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O

Table 3-1 (Cont.) Error Code Profile Configuration

Field Name	Description	Details
Redirect URL	This field determines the redirection URL, this value is populated in LOCATION header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the redirectUrl field for the particular errorScenario is configured appropriately.	DataType: integer Constraints: NA Default Value: NA Presence: O
Retry After	This field determines the value in seconds or particular date after which the service should be retried, this value is populated in Retry-After header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the retry-after field for the particular errorScenario is configured appropriately in seconds.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Description	This field determines the description that needs to be populated in ProblemDetails (Detail field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: O

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.2.3 Peer Configuration

This URI is used to add or update the list of peers wherein each peer consists of ID, host, port or virtualHost, and apiPrefix. The ID of each peer is mapped to Peer Identifier in Peer Set Configuration. The default value is null.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Egress Gateway**.
2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
3. Click the **Peer Configuration** option under **Egress Gateway** to configure peers. The **Peer Configuration** page is displayed.
4. Click **Add** to add the peer configuration. The **Create Peer Configuration** page is displayed.
5. Configure the following fields in the **Create Peer Configuration** page:
 - a. **ID:** Enter a unique Peer identifier.
 - b. **Host:** Enter the Host details of a local peer. It can be IPv4, IPv6, and FQDN details.
 - c. **Port:** Enter the port details of the local host peer.

- d. **API Prefix:** Enter the API prefix details of a peer.
Note: It is recommended to set the value as /.
 - e. **Virtual Host:** Enter the Host details of a remote peer. This FQDN is sent to an alternate route service.
 - f. **healthApiPath:** Parameter to support SCP health check API. It contains path of the health API
6. Click **Save** on the **Create Peer Configuration** page to save the details. Click **Cancel** to discard your progress and go back to **Peer Configuration** page.

① Note

- Use **Edit** icon available in the next column of the specific entry to update configured the Peer Configuration information.
- Use **Refresh** icon to refresh the list of peers configured.

3.2.4 Peer Set Configuration

This URI is used to add or update the list of peer sets wherein each peer set consists of id and list of http/https instances. Each instance consists of priority and peer identifier that is mapped to id in peerconfiguration resource. The id of each peer set is mapped to peerSetIdentifier in routesconfiguration resource. The default value is null.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Egress Gateway**.
2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
3. Click the **Peer Set Configuration** option under **Egress Gateway** to configure peers. The **Peer Set Configuration** page is displayed.
4. Click **Add** to add the peer set configuration. The **Create Peer Set Configuration** page is displayed.
5. Configure the following fields in the **Create Peer Configuration** page:
 - a. **ID:** Enter a unique Peer set identifier.
 - b. Click **Add** in **HTTP Configuration** section to add HTTP Configuration. The **Add HTTP Configuration** dialog box is displayed.
 - i. Enter the following information on this page:
 - i. **Priority:** Enter the Priority of peer to be used in a peer set.
 - ii. **Peer Identifier:** Enter the Peer identifier is the value of peer configured during [Peer Configuration](#).
 - ii. Click **Save** to save HTTP Configuration. Click **Cancel** to discard your progress, close the dialog box, and go back to **Create Peer Set Configuration** page.
 - c. Click **Add** in **HTTPS Configuration** section to add HTTPS Configuration. The **Add HTTPS Configuration** page is displayed.
 - i. Enter the following information on this page:
 - i. **Priority:** Enter the Priority of peer to be used in a peer set.

- ii. **Peer Identifier:** Enter the Peer identifier is the value of peer configured during [Peer Configuration](#).
 - ii. Click **Save** to save HTTP Configuration. Click **Cancel** to discard your progress, close the dialog box, and go back to **Create Peer Set Configuration** page.
6. Click **Save** on the **Create Peer Set Configuration** page to save the details. Click **Cancel** to discard your progress and go back to **Peer Set Configuration** page.

Note

- Use **Edit** icon available in the next column of the specific entry to update configured the Peer Set Configuration information.
- Use **Refresh** icon to refresh the list of peer sets configured.

3.2.5 Routes Configuration

The configuration of "routesconfiguration" is required for Server Header and Overload control feature to map route ID and its corresponding route-level configuration. By default, this configuration is null.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Ingress Gateway** or **Egress Gateway**
2. For Ingress Gateway only, from the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click the **Routes Configuration** option.
The **Routes Configuration** page is displayed.
4. Click **Add** to add the configuration.
The **Create Routes Configuration** page is displayed.
5. Configure the following fields in the **Create Routes Configuration** page.
 - a. **ID:** Value of "id" attribute defines a specific service for route configuration. It specifies the route IDs for which you need to define server header.

Note

Ensure that route ID used in REST configuration is same as the route ID used in values.yaml file

- b. **URI:** Enter URL, or leave the existing URL with existing value.
- c. **Order:** Provide the order of the execution of this route.
- d. **FailureReqCountErrorCodeSeriesId:** Indicates the ID that is used to map the service with the error code ID defined in errorcodeserieslist.
- e. **Filters:** Click **Add** in the **Filters** section and configure the following fields in **Add Filters** window:
 - **args:** Click **Add** in the **args** section and configure the following fields in **Add args** window:

- **Name:** Enter the args name.
- **Value:** Enter the args value.
- Click **Save** on the **Add args** window to save the details. Click **Cancel** to discard your progress and go back to **Add Filters** window.
- **name:** Enter the filter name.
- Click **Save** on the **Add Filters** page to save the details. Click **Cancel** to discard your progress and go back to **Create Routes Configuration** page.
- f. **metadata:** Click **Add** in the right side column to add metadata.
- g. **predicates:** Click **Add** in the right side column to add predicates. **Add predicates** window is displayed.
 - Configure the following fields in the **Add predicates** window:
 - i. **pattern:** Enter pattern details.
 - ii. **Name:** Enter name of the predicate.
 - Click **Save** to save predicates configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.
- h. **rateLimiting:** Click **Add** in the right side column of **methods** section to add methods. **Add methods** window is displayed.
 - Configure the following fields in the **Add methods** window:
 - i. **Name:** Enter the method name.
 - ii. **Rate:** Enter the method rate.
 - iii. **rateLimitPolicy:** Enter the rateLimitPolicy name.
 - Click **Save** to save methods configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.
- i. **serverHeaderDetails:** Configure the following fields in **serverHeaderDetails** section:
 - i. **Enabled:** Use the switch to enable or disable server header at route level.
 - ii. **Error Code Series Id:** : Specify the error list ID.

Note

Ensure that an errorCodeSeries exists corresponding to the errorCodeSeriesId.

6. Click **Save** to save Routes configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.

3.2.6 SBI Error Action Sets

This URI is used to list or update SBI error action sets configuration at Egress gateway. By default this configuration is disabled.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Egress Gateway**
2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
3. Click the **SBI Error Action Sets** option under **Egress Gateway** to configure peers.

The **SBI Error Action Sets** page is displayed.

4. Click **Add** to add an error action set configuration.
The **Create SBI Error Action Sets** page is displayed.
5. Configure the following fields in the **Create SBI Error Action Sets** page:
 - a. **ID**: Enter an unique ID for SBI routing error action set.
 - b. **Action**: The action that needs to be taken when specific criteria set is matched.
 - c. **Attempts**: Maximum number of retries to either same or different peer in case of error or failures from backend.
 - d. **Black List**: Perform the following configuration:
 - i. **Enabled**: This flag enables the peer blacklist feature using the server headers received in the response.
 - ii. **Duration**: The duration for which the peer is blacklisted and no traffic is routed to that peer for this period.
6. Click **Save** on the **Create SBI Error Action Sets** page to save the details. Click **Cancel** to discard your progress and go back to **SBI Error Action Sets** page.

Note

- Use **Edit** icon available in the next column of the specific entry to update configured the **SBI Error Action Sets** information.
- Use **Refresh** icon to refresh the configuration.

3.2.7 SBI Error Criteria Sets

This URI is used to list or update SBI error criteria sets configuration at Egress Gateway. By default, this configuration is disabled.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Egress Gateway**
2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
3. Click the **SBI Error Criteria Sets** option under **Egress Gateway** to configure peers.
The **SBI Error Criteria Sets** page is displayed.
4. Click **Add** to add an error action set configuration.
The **Create SBI Error Criteria Sets** page is displayed.
5. Configure the following fields in the **Create SBI Error Criteria Sets** page:
 - a. **ID**: Enter an unique ID for SBI routing error action set.
 - b. **Method**: Indicate the action that needs to be taken when specific criteria set is matched.
 - c. **Exceptions**: Enter the maximum number of retries to either same or different peer in case of error or failures from backend.
 - d. **Response**: Configure the following fields under **Statuses** section. Click **Add** to add HTTP status details:
 - i. **Status Series**: Enter the HTTP status series for which reroute or retry is triggered, when the error response is received from downstream.

- ii. **Status:** Specify HTTP statuses that belongs to above mentioned status series for which reroute or retry is triggered. To enable retry or reroute for all the HTTP status belonging to a status series, configure this as -1.
 - iii. Click **Save** in **Add Statuses** window to save statuses configuration. Click **Cancel** to discard your progress and go back to **Create SBI Error Criteria Sets** page.
- e. **Cause:** Configure the following field under **Cause** section.
 - i. **ignoreCauseIfMissing:** Enable the switch to ignore cause if it is missing.
 - ii. **Path:** Enter path of the cause.
 - iii. **Reason:** Enter reason of the cause.
- 6. **Headers Matching Script:** Enter headers matching script.
- 7. Click **Save** on the **Create SBI Error Criteria Sets** page to save the details. Click **Cancel** to discard your progress and go back to **SBI Error Criteria Sets** page.
- 8. Click **Refresh** to reload the **SBI Error Criteria Sets** page.

3.2.8 User Agent Header Generation

This URI is used to Enable or Disable User-Agent Header.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Egress Gateway**
2. Click the **Egress Gateway** option to configure the Egress Gateway APIs.
3. Click the **User Agent Header Generation** option under **Egress Gateway** to configure peers.
The **User Agent Header Generation** page is displayed with default configured values.
4. Click **Edit** to update the User Agent Header Generation.
The **Edit User Agent Header Generation** page is displayed.
5. Configure the following fields in the **Edit User Agent Header Generation** page:
 - a. **Enabled:** Use the switch to enable or disable User Agent Header feature.
 - b. **NF Type:** : Attribute to configure the nfType that is used to generate the User Agent Header.
 - c. **NF Instance ID:** : Indicates the UUID (Instance ID) of the UDR deployment used to generate the User Agent Header.
 - d. **NF FQDN:** : Use the allow or deny User Agent from appending the UDR FQDN information while generating the User Agent Header.
 - e. **Add Fqdn To Header:** : It indicates the number of successful responses after which an unhealthy SCP can be marked as healthy.
 - f. **Overwrite Header:** Use this switch to govern if you want to include the User Agent Header generated at UDR Egress Gateway or forward the User Agent received from service request.

Note

- When User Agent Header is enabled but the header information is missing, then it is picked from the OAuthClient module.
- If the User Agent Header is present in the request towards AMF or NRF, then the value present in the header is overwritten or forwarded based on the **Overwrite Header** switch. If this switch is enabled, then the header is overwritten.

6. Click **Save** on the **Edit User Agent Header Generation** page to save the details. Click **Cancel** to discard your progress and go back to **User Agent Header Generation** page.

Note

- Use **Refresh** icon to refresh the User Agent Header Generation.

3.3 Ingress Gateway

This section describes the Ingress Gateway configurations.

In the left navigation pane of the CNC Console application, click **UDR** and then **Ingress Gateway**.

You can select **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** to view their services.

Using **Ingress Gateway Signaling**, UDR users can configure following services:

- [Configurable Error Codes](#)
- [Discard Policy Mapping](#)
- [Error Code Profile](#)
- [Error Code Series](#)
- [OAuth2 Configurations](#)
- [Overload Control Discard Policy](#)
- [Routes Configuration](#)
- [Route level Rate Limiting](#)
- [Server Header Details](#)
- [User Agent Header Validation](#)
- [Pod Protection](#)
- [Readiness Configuration](#)
- [Controlled Shutdown Error Mapping](#)
- [XFCC Header Validation](#)
- [Pod Protection By Rate Limiting](#)
- [Congestion Level Configuration](#)

Using **Ingress Gateway Provisioning**, UDR users can configure following services:

- [Configurable Error Codes](#)
- [Routes Configuration](#)
- [Route level Rate Limiting](#)
- [OAuth2 Configurations](#)
- [Discard Policy Mapping](#)
- [Error Code Profile](#)
- [Error Code Series](#)
- [Overload Control Discard Policy](#)
- [Readiness Configuration](#)
- [Controlled Shutdown Error Mapping](#)

3.3.1 Configurable Error Codes

Perform the following procedure to configure error codes using the CNC Console:

1. In the CNC Console, click **UDR** and then **Ingress Gateway** or **Egress Gateway**
2. For Ingress Gateway only, from the left navigation menu, navigate to the **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click the **Configurable Error Codes** option.
The **Configurable Error Codes** page is displayed with default values.
4. Click **Edit** to update the **Configurable Error Codes** configuration.
The **Edit Configurable Error Codes** page is displayed.
5. Click **Add** .
The **Add Error Scenarios** page is displayed.
6. Update the configuration details as required and click **Save**.
A confirmation message, "Save successfully." appears.
.
7. Click **Edit** .
The **Edit Error Scenarios** page is displayed.
8. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully." appears.
9. If you do not want to modify any configuration, click **Cancel**.
10. Click **Delete** and click **OK** to delete the selected configuration.
11. Click **Save** on the **Edit Configurable Error Codes** page to save the details. Click **Cancel** to discard your progress and go back to **Configurable Error Codes** page.
12. Click **Refresh** to reload the **Configurable Error Codes** page.

3.3.2 Discard Policy Mapping

This URI can be used to update service names and corresponding policy names for the service which is mapped to "ocDiscardPolicies" based on "policyName" and also to enable or disable the Overload Control feature and the sampling period in overload control. By default, the Overload Control feature is disabled and the sampling period is 6000.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. For Ingress Gateway, from the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click the **Discard Policy Mapping** option.
The **Discard Policy Mapping** page is displayed with default configured values.
4. Click **Edit** to update the Discard Policy Mapping.
The **Edit Discard Policy Mapping** page is displayed.
5. Configure the following fields in the **Edit Discard Policy Mapping** page:
 - a. **Enabled**: Use the switch to enable or disable discard policy mapping.
 - b. **Mappings**: Configure the following fields in **Mappings** section. Click **Add** to open **Add Mappings** window and add mapping details:
 - **Service Name**: Enter the service name. This field is used to determine a mapping between service and discard policy name per service name. It must be added in the following format:
<deployment-name>-<servicename>
Note: servicename is fixed and cannot be changed.
 - **Policy Name**: Enter the policy name. It determines a mapping between the service and discards policy name per service.
 - c. Click **Save** on the **Add Mappings** window to save the details. Click **Cancel** to discard your progress and go back to **Edit Discard Policy Mapping** page.

Note

- Use **Edit** icon to edit an existing configuration.
- Use **Delete** icon to remove an existing configuration.

6. **Sampling Period**: Add sampling period. It is the time frame for each cycle of Overload Control per service. Its value is in milliseconds.
7. Click **Save** on the **Edit Discard Policy Mapping** page to save the details. Click **Cancel** to discard your progress and go back to **Discard Policy Mapping** page.

Note

- Use **Refresh** icon to refresh the peer monitoring configuration.

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.3 Error Code Profile

This configuration is used to update the errorcodeprofiles that is used in Overload Control, Controlled Shutdown feature for populating details in error responses when a request is discarded.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Ingress Gateway** or **Egress Gateway**
2. For Ingress Gateway only, from the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click **Error Code Profiles** option.
The **Error Code Profiles** page is displayed with default values for few fields.
4. Click **Add** to add the profiles.
The **Create Error Code Profiles** page is displayed.
5. Configure the fields in the **Create Error Code Profiles** page as mentioned in [Table 3-1](#).
6. Click **Save** on the **Create Error Code Profiles** page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Profiles** page.

Note

- Use the **Edit** icon available in the next column of the specific entry to update the route error profile information.
- Use **Refresh** icon to refresh the route error profile information.

Table 3-2 Error Code Profile Configuration

Field Name	Description	Details
Name	Indicate the error name.	DataType: string Constraints: NA Default Value: NA Presence: M
Error Code	This field determines the HttpStatusCode that needs to be populated in ProblemDetails (HttpStatus field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: M
Error Cause	This field determines the error cause that needs to be populated in ProblemDetails (Cause field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Title	This field determines the title that needs to be populated in ProblemDetails (Title field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O

Table 3-2 (Cont.) Error Code Profile Configuration

Field Name	Description	Details
Redirect URL	This field determines the redirection URL, this value is populated in LOCATION header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the redirectUrl field for the particular errorScenario is configured appropriately.	DataType: integer Constraints: NA Default Value: NA Presence: O
Retry After	This field determines the value in seconds or particular date after which the service should be retried, this value is populated in Retry-After header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the retry-after field for the particular errorScenario is configured appropriately in seconds.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Description	This field determines the description that needs to be populated in ProblemDetails (Detail field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: O

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.4 Error Code Series

This configuration is used to map errorCodeSeries id and the corresponding errorCodeSeries configuration. By default this configuration is null.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. For Ingress Gateway, from the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click **Error Code Series** option.
The **Error Code Series** page is displayed.
4. Click **Add** to add the error code series.
The **Create Error Code Series** page is displayed.
5. Configure the fields in the **Create Error Code Series** page as mentioned in [Table 3-3](#).
6. Click **Add** in the **Error Code Series** section.
The **Add Error Code Series** page is displayed.
7. Configure the fields in the **Add Error Code Series** page as mentioned in [Table 3-4](#).

8. Click **Save** on the **Add Error Code Series** page to save the details. Click **Cancel** to discard your progress and go back to **Create Error Code Series** page.
9. Click **Save** on the **Create Error Code Series** page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Series** page.

Note

- Use the **Edit** icon or **Delete** available in the next column of the specific entry to update or delete the error code series information.
- Use **Refresh** icon to refresh the error code series information.

Table 3-3 Error Code Series

Field Name	Description	Details
ID	Indicates the error code identifier.	DataType: string Constraints: NA Default Value: NA Presence: M
Exception List	Lists the configurable exception or error for an error scenario in Ingress Gateway. The only supported values are: ConnectionTimeout, RequestTimeout, UnknownHostException, ConnectException, RejectedExecutionException, InternalError and NotFoundException, ClosedChannelException, BlackListIpException	DataType: string Constraints: NA Default Value: NA Presence: M
Error Code Series	Lists the error codes for a specific service. Note: "ErrorCodeSeries" is configured only if a set of error responses with specific error codes is expected in server header. If it is not configured then all the error responses will have server header.	DataType: string Constraints: NA Default Value: NA Presence: M

Table 3-4 Error Code Series Configuration

Field Name	Description	Details
Error Set	Possible values for "errorSet" attribute: 5xx, 4xx, 3xx, 2xx, 1xx	DataType: string Constraints: NA Default Value: NA Presence: M
Error Codes	Possible values include all error codes in the respective HttpSeries value assigned for "errorSet". Note: Use single value of "-1" if all error codes in that HttpSeries are to be considered.	DataType: string Constraints: NA Default Value: NA Presence: M

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.5 OAuth2 Configurations

This REST API configuration is required for enabling access token validation using NRF Instance ID and key-ID (K-ID).

Before this configuration, perform the prerequisite steps and helm configuration explained in [OAuth2.0](#).

After Helm configuration, send the REST requests to use configured public key certificates. Using REST-based configuration, you can distinguish between the certificates configured on different NRFs and can use these certificates to validate the token received from a specific NRF.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. For Ingress Gateway, from the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click the **OAuth2 Configurations** option.
The **OAuth2 Configurations** page is displayed.
4. Click **Edit** to add the OAuth2 Configurations.
The **Edit OAuth Signaling** page or **Edit OAuth Provisioning** is displayed.
for Ingress Gateway Signaling and Ingress Gateway Provisioning respectively.
5. Configure the following fields in the **Edit OAuth Signaling** page or **Edit OAuth Provisioning** page:
 - a. **Key ID List:** Click **Add** in the **Key ID List** section and configure the following fields in **Add Key ID List** window:
 - **Key ID:** Enter the Key-ID.
 - **Kubernetes Secret Name:** Enter Kubernetes Secret Name.
 - **Kubernetes Secret Key For Certificate:** Enter Kubernetes Secret Key for the certificate.
 - **Access Token Generation Algorithm:** Enter the Access Token Generation Algorithm.
 - b. Click **Save** on the **Add Key ID List** page to save the details. Click **Cancel** to discard your progress, close the window, and go back to **Edit OAuth Signaling or Provisioning** page.
 - c. **Instance ID List:** Click **Add** in the **Instance ID List** section and configure the following fields in **Add Instance ID List** window:
 - **Instance ID:** Enter the NRF Instance ID.
 - **Kubernetes Secret Name:** Enter the Kubernetes Secret Name.
 - **Kubernetes Secret Key For Certificate:** Enter Kubernetes Secret Key for the certificate.
 - **Access Token Generation Algorithm:** Enter the Access Token Generation Algorithm.
6. Click **Save** on the **Add Instance ID List** page to save the details. Click **Cancel** to discard your progress, close the window, and go back to **Edit OAuth Signaling or Provisioning** page.

7. **Access Token Validation Mode:** Enter the mode of validation, which are INSTANCEID_ONLY, KID_ONLY, or KID_PREFERRED. It will check for keyIdList or instanceIdList for validation of token received based on mode selected.
8. Click **Save** on the **Edit OAuth Signaling or Provisioning** page to save the details. Click **Cancel** to discard your progress and go back to **Edit OAuth Signaling or Provisioning** page.

Note

- Use **Edit** icon available in the next column of the specific entry to update the **Key ID List** or **Instance ID List**.
- Use **Refresh** icon to refresh the configuration.

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.6 Overload Control Discard Policy

This configuration is used to update discard policies that are used in overload control to select the appropriate policy from the configured list based on the load level of a particular service. By default, ocdiscardpolicies is null.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
The **Overload Control Discard Policy** page is displayed.
3. Click **Add** to add the configuration.
The **Create Overload Control Discard Policy** page is displayed.
4. Configure the fields in the **Create Overload Control Discard Policy** page as mentioned in [Table 3-5](#).
5. Click **Add** in the **Policies** section.
The **Add Policies** page is displayed.
6. Configure the fields under the **Add Policies** page as mentioned in [Table 3-6](#).
7. Click **Save** on the **Add Policies** page to save the details. Click **Cancel** to discard your progress and go back to **Create Overload Control Discard Policy** page
8. Click **Save** on the **Create Overload Control Discard Policy** page to save the details. Click **Cancel** to discard your progress and go back to **Overload Control Discard Policy** page

Note

- Use the **Edit** icon available in the next column of the specific entry to update the policy information.
- Use **Refresh** icon to refresh the policy information.

Table 3-5 Discard Policy Configuration

Attribute Name	Description	Details
Name	Name of the discarded policy. Note: name must be the value configured in policyName under ocpolicymapping.	DataType: string Constraints: NA Default Value: NA Presence: M
Scheme	Discarded policy scheme based on percentage.	DataType: string Constraints: NA Default Value: NA Presence: M

Table 3-6 Policies

Attribute Name	Description	Details
Value	Value of priority above which requests are considered as potential candidates for drop. Percentage of requests to drop in the current sampling period over the calculated rate in the previous sampling period.	DataType: string Constraints: NA Default Value: NA Presence: M
Action	Defines the action to be taken on selected requests rejection based on error code.	DataType: string The value can be: RejectWithErrorCode Constraints: NA Default Value: NA Presence: M
Level	Defines the overload level.	DataType: string Constraints: NA Default Value: NA Presence: M
Error Code Profile	Define the error code profiles.	DataType: string Constraints: NA Default Value: NA Presence: M

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.7 Routes Configuration

The configuration of "routesconfiguration" is required for Server Header and Overload control feature to map route ID and its corresponding route-level configuration. By default, this configuration is null.

Perform the following configurations:

1. In the CNC Console application, click **UDR** and then **Ingress Gateway** or **Egress Gateway**
2. For Ingress Gateway only, from the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.

3. Click the **Routes Configuration** option.
The **Routes Configuration** page is displayed.
4. Click **Add** to add the configuration.
The **Create Routes Configuration** page is displayed.
5. Configure the following fields in the **Create Routes Configuration** page.
 - a. **ID**: Value of "id" attribute defines a specific service for route configuration. It specifies the route IDs for which you need to define server header.

Note

Ensure that route ID used in REST configuration is same as the route ID used in values.yaml file

- b. **URI**: Enter URL, or leave the existing URL with existing value.
- c. **Order**: Provide the order of the execution of this route.
- d. **FailureReqCountErrorCodeSeriesId**: Indicates the ID that is used to map the service with the error code ID defined in errorcodeserieslist.
- e. **Filters**: Click **Add** in the **Filters** section and configure the following fields in **Add Filters** window:
 - **args**: Click **Add** in the **args** section and configure the following fields in **Add args** window:
 - **Name**: Enter the args name.
 - **Value**: Enter the args value.
 - Click **Save** on the **Add args** window to save the details. Click **Cancel** to discard your progress and go back to **Add Filters** window.
 - **name**: Enter the filter name.
 - Click **Save** on the **Add Filters** page to save the details. Click **Cancel** to discard your progress and go back to **Create Routes Configuration** page.
- f. **metadata**: Click **Add** in the right side column to add metadata.
- g. **predicates**: Click **Add** in the right side column to add predicates.
Add predicates window is displayed.
 - Configure the following fields in the **Add predicates** window:
 - i. **pattern**: Enter pattern details.
 - ii. **Name**: Enter name of the predicate.
 - Click **Save** to save predicates configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.
- h. **rateLimiting**: Click **Add** in the right side column of **methods** section to add methods.
Add methods window is displayed.
 - Configure the following fields in the **Add methods** window:
 - i. **Name**: Enter the method name.
 - ii. **Rate**: Enter the method rate.
 - iii. **rateLimitPolicy**: Enter the rateLimitPolicy name.

- Click **Save** to save methods configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.
- i. **serverHeaderDetails**: Configure the following fields in **serverHeaderDetails** section:
 - i. **Enabled**: Use the switch to enable or disable server header at route level.
 - ii. **Error Code Series Id** : Specify the error list ID.

Note

Ensure that an errorCodeSeries exists corresponding to the errorCodeSeriesId.

6. Click **Save** to save Routes configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.

3.3.8 Route Level Rate Limiting

This section describes how to configure Route Level Rate Limiting using the CNC Console.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click the **Route Level Rate Limiting** option.
The **Route Level Rate Limiting** page is displayed.
4. Click **Edit** to update the Route Level Rate Limiting.
The **Edit Route Level Rate Limiting** page is displayed.
5. Configure the following fields in the **Edit Route Level Rate Limiting** page:
 - a. **Enabled**: Use the switch to enable or disable Route Level Rate Limiting.
 - b. **Sampling Period**: Add sampling period.
 - c. **Rate Limit Policies**: Configure the following fields in **Rate Limit Policies** section.
Click **Add** to open **Add Rate Limit Policies** window and add Rate Limit Policies details:
 - **Name**: Enter the rate limit policy name.
 - **Value**: Value of priority above which requests are considered as potential candidates for drop. Percentage of requests to drop in the current sampling period over the calculated rate in the previous sampling period..
 - **Action**: Defines the action to be taken on selected requests rejection based on error code.
 - **Scheme**: Rate limit policy scheme based on percentage.
 - **Error Code Profile**: Define the error code profiles.
 - d. Click **Save** on the **Add Rate Limit Policies** window to save the details. Click **Cancel** to discard your progress and go back to **Edit Route Level Rate Limiting** page.

Note

- Use **Edit** icon to edit an existing configuration.
- Use **Delete** icon to remove an existing configuration.

6. Click **Save** on the **Edit Route Level Rate Limiting** page to save the details. Click **Cancel** to discard your progress and go back to **Route Level Rate Limiting** page.

Note

- Use **Refresh** icon to refresh the Route Level Rate Limiting.

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.9 Server Header Details

This API can be used for adding Server Header in the error responses sent from Ingress Gateway. By default, this feature is disabled. To enable the feature, invoke the following REST API and update the enable switch.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling**.
3. Click the **Server Header Details** option.
The **Server Header Details** page is displayed with default configured values.
4. Click **Edit** to update the Server Header Details.
The **Edit Server Header Details** page is displayed.
5. Configure the following fields in the **Edit Server Header Details** page:
 - a. **Enabled**: Use the switch to enable or disable Server Header.
 - b. **Error Code Series Id**: Specify the error list ID.

Note

Ensure that an [Error Code Series](#) exists corresponding to the errorCodeSeriesId.

- c. **Configuration**: Configure the following fields in **Configuration** section:
 - **NF Type**: Specify the type of network function. In this case, it is UDR.
 - **NF Instance Id**:: Enter the UDR instance ID.
6. Click **Save** on the **Edit Server Header Details** page to save the details. Click **Cancel** to discard your progress and go back to **Server Header Details** page.

Note

- Use **Refresh** icon to refresh the Server Header Details.

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.10 User Agent Header Validation

This section describes how to configure User Agent Header Validation using the CNC Console.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling**.
3. Click the **User Agent Header Validation** option under **Ingress Gateway Signaling** to configure peers.
The **User Agent Header Validation** page is displayed with default configured values.
4. Click **Edit** to update the User Agent Header Validation.
The **Edit User Agent Header Validation** page is displayed.
5. Configure the following fields in the **Edit User Agent Header Validation** page:
 - a. **Enabled**: Use the switch to enable or disable User Agent Header feature.
 - b. **Validation Type**: Add the required validation type.
 - c. **Consumer NF Types**: Attribute to configure the consumer nfType.
6. Click **Save** on the **Edit User Agent Header Validation** page to save the details. Click **Cancel** to discard your progress and go back to **User Agent Header Validation** page.

Note

- Use **Refresh** icon to refresh the User Agent Header Validation.

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.11 Pod Protection

This configuration is used to configure the pod protection for Ingress Gateway pod.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling** option to configure the Ingress Gateway Signaling APIs.
3. Click the **Pod Protection** option under **Ingress Gateway Signaling**.
The **Pod Protection** page is displayed.
4. Click **Edit** to edit the configuration.

The **Edit Pod Protection** page is displayed.

5. Configure the fields in the **Edit Pod Protection** page as mentioned in [Table 3-7](#).
6. Configure the fields under the **Congestion Control Configuration** section as mentioned in [Table 3-8](#).
7. Click **Add** in the **States** section to configure different states for Pod Protection.
The **Add States** page is displayed.
8. Configure the fields in the **Add States** page as mentioned in [Table 3-9](#).
9. Configure the fields in the **Congestion Control Threshold** section in the **Add States** page as mentioned in [Table 3-10](#).
10. Click **Add** in the **Error Action** section the **Add States** page.
The **Add Error Action** page is displayed.
11. Configure the fields in the **Add Error Action** page as mentioned in [Table 3-11](#).
12. Click **Add** in the **Entry Action** section the **Add Entry Action** page.
The **Add Entry Action** page is displayed.
13. Configure the fields in the **Add Entry Action** page as mentioned in [Table 3-12](#).
14. Click **Save** on the **Add Error Action** page to save the details. Click **Cancel** to discard your progress and go back to **Error Action** page.
15. Click **Save** on the **Add States** page to save the details. Click **Cancel** to discard your progress and go back to **Edit Pod Protection** page.
16. Click **Save** on the **Edit Pod Protection** page to save the details. Click **Cancel** to discard your progress and go back to **Pod Protection** page.

Note

- Use the **Edit** icon available in the next column of the specific entry to update or delete the states information.
- Use **Refresh** icon to refresh the states information.

Table 3-7 Pod Protection Options

Field Name	Description	Details
Enabled	This attribute indicates if the Pod Protection feature is enabled or disabled.	DataType: Boolean Constraints: true, false Default Value: false
Monitoring Interval	This attribute indicates the periodicity at which the overload state is monitored. Unit: Milliseconds Note: The proposed value for this attribute is 100.	DataType: Integer Constraints: NA Default Value: NA

Table 3-8 Congestion Control Configurations

Field Name	Description	Details
Enabled	This attribute allows the configuration of pod protection attributes for the Ingress Gateway pods. Note: This must be set to true for Pod Protection feature.	DataType: Boolean Constraints: true, false Default Value: false
State Change Sample Count	This attribute indicates the number of times the pod must be in the particular congestion state before transitioning to another state. Note: The proposed value for this attribute is 10.	DataType: Integer Constraints: NA Default Value: NA
Action Sampling Period	This attribute indicates the interval at which the configured action must be considered. The actions are configured under <code>entryAction.action</code> attribute. The interval is calculated as $(\text{actionSamplingPeriod} * \text{monitoringInterval})$. Note: The proposed value for this attribute is 3.	DataType: Integer Constraints: NA Default Value: NA

Table 3-9 States

Field Name	Description	Details
Name	The name of the congestion state. - Normal: The pod is not in overload state. - Danger of Congestion (Doc): The pod is about to go into the congested state. Actions configured in the <code>entryAction</code> is performed. - Congested state: The pod is in congested state. Actions configured in the <code>entryAction</code> is performed.	DataType: String Constraints: Normal, Doc, Congested Default Value: NA
Weight	The weight of the congestion state. The weight indicates the critical of the congestion state. The lower the value, the lower the criticality.	DataType: Integer Constraints: NA Default Value: NA

Table 3-10 Congestion Control Threshold

Field Name	Description	Details
CPU	The CPU threshold is expressed in percentage.	DataType: Integer Constraints: NA Default Value: NA
Pending Message	The number of pending messages to be processed, expressed in absolute count. Note: The recommended value for this attribute is for DoC is 1500 and for Congested is 2000.	DataType: Integer Constraints: NA Default Value: NA

Table 3-11 Entry Action

Field Name	Description	Details
Action	This attribute indicates the action for the congestion state. - AcceptIncomingConnections: The action indicates whether the incoming new connection is accepted or rejected based on the overload state. - MaxConcurrentStreamsUpdate: The action indicates whether to increase or decrease the max concurrent stream for all incoming connections till the maxConcurrentStreamsPerCon is reached.	DataType: String Constraints: MaxConcurrentStreamsUpdate, AcceptIncomingConnections Default Value: NA

Table 3-12 Possible Arguments

Field Name	Description	Details
Accept	The attribute indicates if the incoming connection should be accepted or not. Applicable when the action is AcceptIncomingConnections. true: The incoming connection is accepted. false: The incoming connection is rejected. The proposed value for Normal state is true, and for DoC and Congested state is false.	DataType: Boolean Constraints: true, false Default Value: NA
Increment by	The attribute indicates the factor by which the current concurrent streams value will be incremented till it reaches maxConcurrentStreamsPerCon. Note: The proposed value for this attribute is 30 for Normal and DoC state.	DataType: Integer Constraints: NA Default Value: NA
Increment by Action Sampling Period	The attribute indicates the time interval at which the incrementBy is applied to reach maxConcurrentStreamsPerCon. If not provided, the actionSamplingPeriod is used. Note: The proposed value for this attribute is 3 for Normal and DoC state. Unit is seconds.	DataType: Integer Constraints: NA Default Value: NA
Decrement by	The attribute indicates the factor by which the current concurrent streams value will be decremented till it reaches maxConcurrentStreamsPerCon. Note: The proposed value for this attribute is 30 for DoC and Congested state.	DataType: Integer Constraints: NA Default Value: NA
Decrement by Action Sampling Period	The attribute indicates the time interval at which the decrementBy is applied to reach maxConcurrentStreamsPerCon. If not provided, the actionSamplingPeriod is used. Note: The proposed value for this attribute is 1 for DoC and Congested state. Unit is seconds.	DataType: Integer Constraints: NA Default Value: NA
Max Concurrent Streams Per Con	The attribute indicates the maximum number of concurrent streams per connection allowed.	DataType: Integer Constraints: NA Default Value: NA

Note

- Use **Refresh** icon to refresh the Pod Protection.

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.12 Readiness Configuration

This section describes how to configure Readiness Configuration using the CNC Console.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click the **Readiness Configuration** option.
The **Readiness Configuration** page is displayed.
4. Click **Edit** to add the configuration.
The **Edit Readiness Configuration** page is displayed.
5. Click **Add** to add the configuration.
The **Add Service Profiles** page is displayed.
6. Update the configuration details as required in the **Add Service Profiles** page.
7. Click **Save** to save the configuration. Click **Cancel** to discard your progress, close the window, and go back to **Edit Readiness Configuration** page.
8. Click **Save** to save Readiness Configuration. Click **Cancel** to discard your progress and go back to **Readiness Configuration** page.

3.3.13 Controlled Shutdown Error Mapping

This configuration is used for mapping between the routes and the error code profile that is used when the Ingress Gateway rejects incoming requests.

Perform the following configurations:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling** or **Ingress Gateway Provisioning** option to configure the Ingress Gateway Signaling or Provisioning APIs.
3. Click **Controlled Shutdown Error Mapping** option.
The **Controlled Shutdown Error Mapping** page is displayed.
4. Click **Edit** to edit the configuration.
The **Edit Controlled Shutdown Error Mapping** page is displayed.
5. Click **Add** in the **Edit Controlled Shutdown Error Mapping** section.
The **Add Route Error Profile List** page is displayed.
6. Configure the fields in the **Add Route Error Profile List** page as mentioned in [Table 3-13](#).

7. Click **Save** on the **Add Route Error Profile List** page to save the details. Click **Cancel** to discard your progress and go back to **Edit Controlled Shutdown Error Mapping** page.
8. Click **Save** to save Controlled Shutdown Error Mapping. Click **Cancel** to discard your progress and go back to **Controlled Shutdown Error Mapping** page.

Note

- Use **Refresh** icon to refresh the Route Level Rate Limiting.

Table 3-13 Controlled Shutdown Error Mapping

Field Name	Description	Details
Route ID	The route id that is configured in routes configuration.	DataType: string Constraints: NA Possible values: <subscription_mapping, accesstoken_mapping, registration_mapping, disc_mapping> The possible values should match with the routeld of the routes configuration. Default Value: NA
Error Profile Name	The error profile name that is used to fetch the error from errorcodeprofiles.	DataType: string Constraints: NA Default Value: shutdownerror503

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.14 XFCC Header Validation

Perform the following procedure to XFCC Header Validation using the CNC Console:

1. From the left navigation menu, navigate to **UDR** and click the **Ingress Gateway** option.
2. From the left navigation menu, navigate to **Ingress Gateway Signaling** option to configure the Ingress Gateway Signaling APIs.
3. Click the **XFCC Header Validation** option.
The **XFCC Header Validation** page is displayed with default values.
4. Click **Edit** to update the **XFCC Header Validation** configuration.
The **Edit XFCC Header Validation** page is displayed.
5. Click **Add**.
The **Add PeerList** page is displayed.
6. Update the configuration details as required and click **Save**.
7. Click **Edit**.
The **Edit PeerList** page is displayed.
8. Update the configuration details as required and click **Save**.
9. If you do not want to modify any configuration, click **Cancel**.
10. Click **Delete** and click **OK** to delete the selected configuration.

11. Click **Save** on the **Edit XFCC Header Validation** page to save the details. Click **Cancel** to discard your progress and go back to **XFCC Header Validation** page.
12. Click **Refresh** to reload the **Configurable Error Codes** page.

For more information about the configuration parameters, see *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.3.15 Congestion Level Configuration

Perform the following procedure to configure the congestion level of the pods.

1. In the CNC Console, click **UDR** and then **Ingress Gateway**
2. From the left navigation menu, navigate to **Ingress Gateway Signaling**.
3. Click the **Congestion Level Configuration** option under Ingress Gateway Signaling. The **Congestion Level Configuration** page is displayed.
4. Click **Edit** to edit the configuration. The **Edit Congestion Level Configuration** page is displayed.
5. Configure the fields in the **Edit Congestion Level Configuration** page as follows:

Table 3-14 Congestion Level Configuration

Field Name	Description	Details
Refresh Interval (seconds)	This attribute indicates the refresh interval for the scheduler to calculate congestion level in milliseconds. CPU levels are stable when tested with values more than 500 ms as a refresh interval. Any value less than this, the CPU spikes are observed for a shorter period of time, and denied requests are processed based on the action defined for that congestion level.	Data Type: Integer Constraints: NA Default Value: 500
Levels	This attribute indicates an array of different levels of congestion.	Data Type: Object Constraints: NA Default Value: NA

6. Click **Add** to add the configuration levels.
7. Configure the fields in the **Add Levels** page as follows:

Table 3-15 Congestion Levels

Attribute Name	Description	Details
Level Name	Indicates the name of the congestion level. The possible values are Normal, Danger of Congestion, and Congested.	Data Type: String Constraints: Normal, Danger of Congestion, Congested Default Value: NA

Table 3-15 (Cont.) Congestion Levels

Attribute Name	Description	Details
Level Value	Indicates numerical values for each congestion level. The value ranges from 1-10. The values should be defined from 1 to 10 in increasing order. The default level of the system is 0. This is the state of the system when the CPU is less than the onset of L1 defined in the configuration.	Data Type: Integer Constraints: 1 -10 Default Value: 0
Resources	Indicates the list of resources.	Data Type: Object Constraints: NA Default Value: NA

8. Click **Add** to add the resource details.
9. Configure the fields in the **Add Resources** page as mentioned in the below table.

Table 3-16 Resources

Attribute Name	Description	Details
Resource Name	Indicates the resource names. The possible value is CPU.	Data Type: String Constraints: NA Default Value: CPU
Onset Threshold	Indicates the onset threshold. Note: Onset must be more than the level of abatement. It is denoted by percentage.	Data Type: Integer Constraints: NA Default Value: NA
Abatement Threshold	Indicates the abatement threshold. It is denoted by percentage. Note: - Abatement must be less than the onset of the same level. - Abatement must be more than the onset of the previous level.	Data Type: Integer Constraints: NA Default Value: NA

10. Click **Save** on the **Add Resources** page to save the details. Click **Cancel** to discard your progress and go back to the **Add Levels** page.
11. Click **Save** on the **Add Levels** page to save the details. Click **Cancel** to discard your progress and go back to **Edit Congestion Level Configuration** page.
12. Click **Save** on the **Congestion Level Configuration** page to save the details.

3.3.16 Pod Protection By Rate Limiting

Perform the following procedure to enable and identify the pods that must be protected by Rate Limiting feature.

1. In the CNC Console, click **UDR** and then **Ingress Gateway**
2. From the left navigation menu, navigate to **Ingress Gateway Signaling**.
3. Click the **Pod Protection By Rate Limiting** option under the **Ingress Gateway Signaling**. The **Pod Protection By Rate Limiting** page is displayed.

4. Click **Edit** to edit the configuration.
The **Edit Pod Protection By Rate Limiting** page is displayed.
5. Configure the fields in the **Edit Pod Protection By Rate Limiting** page as mentioned in the following table.

Table 3-17 Pod Protection By Rate Limiting

Field Name	Description	Details
Enabled	The attribute indicates if rate limiting feature is enabled for the Ingress Gateway pod protection.	Data Type: Boolean Constraints: true or false Default Value: false
Fill Rate	This attribute indicates the number of requests to be processed by an Ingress Gateway pod in one second (1000 ms).	Data Type: Integer Constraints: more than 0 Default Value: 2800 Unit: milliseconds
Routes	This attribute indicates the route levels for the rate limiting.	Data Type: Array (object of Routes) Constraints: NA Default Value: For more information, see Table 3-18 .
Error Code Profile	This attribute indicates the error profile to be sent for the rejected requests.	Data Type: String Constraints: HELM, REST, ERR_RATE_LIMIT Default Value: error429
Denied Request Actions	This attribute indicates the request actions to be performed for the rejected requests.	Data Type: Object Constraints: NA Default Value: For more information, see Table 3-19 .
Priority Header Name	Determines the priority of the request using the configured header. Indicates the name of the header..	Data Type: String Constraints: NA Default Value: 3gpp-sbi-message-priority
Default Priority	This attribute determines the priority of the request.	Data Type: Integer Constraints: 0-31 Default Value: 24

6. Click **Add** in the **Routes** section to configure the different routes for Pod Protection By Rate Limiting feature.
7. The page is displayed.
8. Configure the fields in the **Add Routes** page as follows:

Table 3-18 Routes

Field Name	Description	Details
Id	Indicates the route ID that processes the request. The number of route IDs can be more than 1.	Data Type: Integer Constraints: 1-10 Default Value: 1

Table 3-18 (Cont.) Routes

Field Name	Description	Details
Path	Indicates the path to be matched to apply fill rate on the route.	Data Type: String Constraints: NA Default Value: /nudr-group-id-map/**
Methods	Indicates the HTTP method of the route for which the rate limiting is defined. All valid HTTP methods are allowed, for example: • POST • PUT • PATCH • DELETE • GET	Data Type: String Constraints: • POST • PUT • PATCH • DELETE • GET Default Value: GET
Percentage	This attribute indicates the percentage of requests for a route. If the fill rate is 1000 and routes .percentage is 40%, the route process 400 requests in 1000 ms. The minimum value is 0.1. Note: The total percentage across all routes must not exceed 100.	Data Type: Integer Constraints: 1-100 Default Value: 100
Default Priority	This attribute determines the priority of the request.	Data Type: Integer Constraints: more than 1 Default Value: 24

9. Click **Save** on the **Add Routes** page to save the details. Click **Cancel** to discard your progress and go back to **Edit Pod Protection By Rate Limiting** page.
10. Click **Add** in the **Denied Request Actions** section to configure the actions for rejected requests.
11. The **Add Denied Request Actions** page is displayed.
12. Configure the fields in the **Add Denied Request Actions** page as follows:

Table 3-19 Denied Request Actions

Field Name	Description	Details
Id	Indicates the unique ID of the rejected request.	Data Type: Integer Constraints: 1-10 Default Value: 1,2,3,4
Congestion Level	Indicates the congestion level of the pods. This value depends on the value field in the congestion level configuration or the default value 0. For more information about congestion level configuration, see Congestion Level Configuration .	Data Type: Integer Constraints: 1-10 Default Value: 0,1,2,3,4

Table 3-19 (Cont.) Denied Request Actions

Field Name	Description	Details
Action	Rejects or processes a denied request based on the value of <code>congestionLevel</code> attribute in the <code>deniedRequestActions</code> object. The possible values are REJECT and CONTINUE.	Data Type: String Constraints: REJECT, CONTINUE Default Value: 0- CONTINUE 1- CONTINUE 2- REJECT 3- REJECT
Error Code Profile	Indicates a valid error code for the rejected requests based on the action.	Data Type: String Constraints: Any error code profile that is configured using <code>errorcodeprofiles</code> API Default Value: error429

13. Click **Save** on the **Add Denied Request Actions** page to save the details. Click **Cancel** to discard your progress and go back to **Edit Pod Protection By Rate Limiting** page.
14. Click **Save** on the **Edit Pod Protection By Rate Limiting** page to save the details. Click **Cancel** to discard your progress and go back to **Pod Protection By Rate Limiting** page.

3.4 cnDBTier

Perform the following procedure to view the cnDBTier version, the status of cnDBTier clusters, georeplication status, and georeplication recovery status on the CNC Console.

Note

The following cnDBTier APIs are read only.

3.4.1 cnDBTier Health

Perform the following procedure to view the health status of the microservices:

1. From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
2. Click **cnDBTier Health** to view the health status of the databases and their services. The **cnDBTier Health** page is displayed.
 - Click the **Backup Manager Health Status** to view the health status of the backup manager.

The **Backup Manager Health Status** page is displayed.

Note

The following APIs are read-only.

Table 3-20 Backup Manager Health Status

Fields	Description
Service Name	This attribute displays the name of the service.
Service Status	This attribute displays the status of the service. Possible values are UP, DOWN.
DB Connection Status	This attribute displays the database connection status. Possible values are UP, DOWN.
Overall Backup Manager Service Health	This attribute displays the overall health status of the backup manager service. Possible values are UP, DOWN.
Backup Executor Health Status	This attribute displays the details of the backup executor.
Node Id	This attribute displays the id of the node.
DB Connection Status	This attribute displays the database connection status with the nodes. Possible values are UP, DOWN.

- Click the **Monitor Health Status** to view the health status of the services. The **Monitor Health Status** page is displayed.

 Note

The following APIs are read-only.

Table 3-21 Monitor Health Status details

Attribute	Description
Service Name	This attribute displays the database service name.
DB Connection Status	This attribute displays the database connection status. Possible values are UP, DOWN.
Metric Scrape Status	This attribute displays the status of the metric scrape, that is if the metrics are fetched or not. If the metrics are fetched then the service is up and vice versa. Possible values are UP, DOWN.
Overall Monitor Service Health	This attribute displays the overall health status of the monitor service. Possible values are UP, DOWN.

- Click the **NDB Health Status** to view the health status of the network database. The **NDB Health Status** page is displayed.

 Note

The following APIs are read-only.

Table 3-22 NDB Health Status details

Attribute	Description
Local Site Name	This attribute displays the name of the local site.
NDB Health Status Details	This attribute displays the health status of the network database.
Service Name	This attribute displays the name of the service.
Service Status	This attribute displays the status of the service. Possible values are UP, DOWN.
PVC Health Status	This attribute displays the health status of the PVC. Possible values are UP, DOWN, and NA. Note: This attribute is set to NA when some of the database pods are not connected to the PVC.

- Click the **Replication Health Status** to view the health status of the replication sites. The **Replication Health Status** page is displayed.

 Note

The following APIs are read-only.

Table 3-23 Replication Health Status details

Attribute	Description
Local Site Name	This attribute displays the name of the local site.
Health Status Details	This attribute displays the health status details. The number of rows in this table varies depending on the type of deployment (for example, two-site, three-site deployments).
Service Name	This attribute displays the name of the available service.
Service Status	This attribute displays the status of the available service. Possible values are UP, DOWN.
DB Connection Status	This attribute displays the database connection status. Possible values are UP, DOWN.
Overall Replication Service Health	This attribute displays the overall health status of the replication service. Possible values are UP, DOWN.

3.4.2 Backup List

Perform the following procedure to view the list of completed backups:

- From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
- Click the **Backup List** to create and check the status of on-demand backups in the cnDBTier.

Table 3-24 Backup List

Fields	Description
Site Name	This field displays the name of the current site to which UDR is connected.
Backup Details	This attribute displays the following information such as backup id, backup size, and backup creation timestamp.
Backup Id	This field displays the ID of the stored backup.
Backup Size (bytes)	This field displays the size of the stored backup.
Creation TimeStamp	This field displays the time recorded when the backup was stored.

3.4.3 cnDBTier Backup Status

Perform the following procedure to view the cnDBTier backup status:

1. From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
2. Click the **cnDBTier Backup Status** to check the status of backups in the cnDBTier.

Table 3-25 cnDBTier Backup Status

Fields	Description
Current Timestamp	This field displays the current date and time.
Is Backup in Progress	This field displays whether the On Demand Backup is in progress or not.
Next Backup Scheduled Timestamp	This field displays the scheduled (date and time) of the next On Demand Backup.

3.4.4 cnDBTier Version

Perform the following procedure to view the version:

1. From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
2. Click the **cnDBTier Version** to view the version.

Table 3-26 cnDBTier Version Attributes

Fields	Description
cnDBTier Version	This field displays the cnDBTier version.
NDB Version	This field displays the network database (NDB) version.

3.4.5 Database Statistics Report

Perform the following procedure to view the available database:

1. From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
2. Click the **Database Statistics Report** to view the available database.

Table 3-27 Database Statistics Report

Fields	Description
Database Count	This field displays the number of available database.
Database Tables Count	This field displays the available database names and their table count.
Database Name	This field displays the database name.
Table Count	This field displays the table count for each database.
Database Table Rows Count	This field displays the table rows present in each table.
Database Name	This field displays the database name.

- a. Click on the **View**. The View Database Table Rows Count page is displayed.

Table 3-28 View Database Table Rows Count

Fields	Description
Database Name	This field displays the database name.
Tables	This field displays the table names and the corresponding rows in each table.
Database Name	This field displays the database name.
Table Name	This field displays the table name.
Row Count	This field displays the table rows present in each table.

3.4.6 Georeplication Recovery

Perform the following procedure to mark cnDBTier cluster as failed, execute georeplication recovery, and monitor their status:

1. From the left navigation menu, navigate to **NRF** and then click **cnDBTier** tab.
2. Click **Georeplication Recovery** to access the Georeplication Recovery Status of the cnDBTier cluster. This includes options such as Update Cluster As Failed, Start Georeplication Recovery, and Georeplication Recovery Status.
 - Click **Update Cluster As Failed** to mark the cnDBTier cluster as **FAILED**. The **Update Cluster As Failed** page is displayed.

Table 3-29 Update Cluster As Failed

Fields	Description
Cluster Names	This field displays a list of cnDBTier clusters that can be marked as failed.
Failed Cluster Names	This field displays the list of cnDBTier clusters that are marked as failed.

Click **Update Cluster**. The selected cnDBTier cluster name is updated in the **Failed Cluster Names** field.

- Click **Start Georeplication Recovery** to start the georeplication recovery process for a failed site. The **Start Georeplication Recovery** page is displayed.

Table 3-30 Start Georeplication Recovery

Attribute	Description
Failed Cluster Name	This field displays a list of all the cnDBTier clusters that have been marked as failed.
Backup Cluster Name (Optional)	This field displays a list of all the healthy cnDBTier clusters. If no cnDBTier cluster is selected, the system uses the first available healthy cnDBTier cluster for the backup.

Click **Start Georeplication Recovery** to initiate georeplication recovery.

- Click **Georeplication Recovery Status** to view the status of georeplication recovery for cnDBTier clusters.
The **Georeplication Recovery Status** page is displayed.

Table 3-31 Georeplication Recovery Status

Attribute	Description
Local Cluster Name	This field displays the name of the local cnDBTier cluster.
Georeplication Recover Status Details	This field displays the details of the georeplication recovery status of cnDBTier clusters.
Cluster Name	This field displays the cnDBTier clusters by name.
Georeplication Recovery Status	This field displays the current georeplication recovery status of the corresponding cnDBTier cluster. For more information about the georeplication recovery status, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i> .

3.4.7 GeoReplication Status

Perform the following procedure to view the local site and remote site name to which UDR is connected:

- From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
- Click the **GeoReplication Status** to view the local site and remote site name to which UDR is connected.

Table 3-32 GeoReplication Status

Fields	Description
Local Site Name	This field displays the local site name to which UDR is connected.
Remote Site Name	This field displays the remote site name.
Replication Status	This field displays the replication status with corresponding sites.
Seconds Behind Remote Site	This field displays the number of seconds that the last record read by the local site is behind the latest record written by the remote site for all the replication groups.

- Click the **View** icon in the **Actions** menu to view the **View Geo Replication Status** screen.

Table 3-33 Geo Replication Status

Fields	Description
Replication Group Delay	This field displays the number of seconds that the last record read by the local site is behind the latest record written by the remote site for individual replication groups.
Replication Channel Group Id	This field displays the ID of the replication channel group.

4. Click the **View** icon to view the **Replication Group Delay** attributes:

Table 3-34 View Replication Group Delay

Fields	Description
Channel Details	This field displays the channel details such as Remote Replication IP and Role.
Remote Replication IP	This field displays the IP of the remote replication channel.
Role	This field displays the role of the replication channel IP.

3.4.8 HeartBeat Status

Perform the following procedure to view the connectivity between local site and remote site name to which SCP is connected:

1. From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
2. Click the **HeartBeat Status** to view the connectivity between local site and remote site name to which UDR is connected.

Table 3-35 HeartBeat Status Details

Fields	Description
Site Name	This field displays the name of the current site to which NRF is connected.
HeartBeat Details	This field displays the following information like remote site name, heart beat status, heart beat lag, and replication channel group id.
Remote Site Name	This field displays the remote site name.
Heartbeat Status	This field displays the connectivity status with corresponding sites.
Heartbeat Lag	This field displays the lag or latency in seconds it took to synchronize between sites.
Replication channel Group Id	This field displays the ID of the replication channel group.

3.4.9 Local Cluster Status

Perform the following procedure to view the local cluster status for the current site:

1. From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
2. Click the **Local Cluster Status** to view the local cluster status for the current site:

Table 3-36 Local Cluster Status

Fields	Description
Site Name	This field displays the name of the current site to which UDR is connected.
Cluster Status	This field displays the local cluster status for the current site.

3.4.10 On Demand Backup

Perform the following procedure to view the available database:

1. From the left navigation pane, click the **UDR** tab, and then click the **cnDBTier** tab. The cnDBTier page is displayed.
2. Click the **On Demand Backup** to view the initiated on-demand backups to store data.

Table 3-37 On Demand Backup Details

Fields	Description
Site Name	This field displays the name of the current site to which UDR is connected.
DR Status	This field displays the status of DR.
Backup Id	This field displays the ID of the stored backup.
Backup Status	This field displays the status of backup.
Remote Transfer Status	The field displays the status of remote transfer.
Initiate Backup	The field displays whether the backup is initiated or not.

- a. Click **Edit**. The **Edit On Demand Backup** page appears.

Note

The **Edit** mode is available only for Initiate Backup.

- b. Use the toggle option to Initiate Backup and click **Save**. A confirmation message, "Save successfully" appears.
- c. Click **Cancel** to navigate back to the On Demand Backup page.
- d. Click **Refresh** to reload the On Demand Backup page.

3.5 Access Token Validation

To validate an access token, go to **UDR** then, **Access Token Validation** in the left navigation pane of the CNC Console application, and then **Signaling** or **Provisioning**. Based on the selected option, the **Access Token Validation** page for either **Signaling** or **Provisioning** appears, respectively.

This page when loads for the first time shows empty value with labels.

To edit this configuration, you are required to set the **Oauth validation** flag as true at the time of NF installation. Otherwise, it sends a Bad Request Response. For more information, see [Oauth2.0](#).

Editing the Access Token Validation Details

To edit the access token details:

1. Click **Edit**. The **Edit Access Token Validation** page appears.
2. To add the instance ID, click **Add**.
 - a. The **Add Instance ID List** dialog box appears.

Figure 3-4 Add Instance ID List

×

Add Instance ID List

Access Token Generation Algorithm:

Kubernetes Secret Key For Certificate:

Instance ID:

Kubernetes Secret Name:

Access Token Generation Algorithm

Kubernetes Secret Key For Certificate

Instance ID

Kubernetes Secret Name

Save

Cancel

- b. Fill in the required details and click **Save** to add the instance ID.
- c. Click **Cancel** to exit the Add Instance ID List dialog box.

Note

Similarly, you can add the Key ID List details.

3. To edit any Instance ID or Key ID detail, click **Edit** corresponding to that particular record under relevant section.
 - a. The **Edit Instance ID List** or **Edit Key ID List** dialog box appears.
 - b. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully." appears.
 - c. Click **Cancel** to access the Edit Access Token Validation page.
4. To delete any Instance ID or Key ID detail, click **Delete** corresponding to that particular record.

- a. A confirmation message appears. Click **OK** to confirm the deletion of record.
- b. Click **Cancel** to exit the delete dialog box and access the **Edit Access Token Validation** page.
5. Update the **Access Token Validation Mode** if required.
6. Click **Save** to update the changes **[OR]** click **Cancel** to exit the Edit Access Token Validation page.
7. Click **Refresh** to reload the Access Token Validation page.

3.6 Subscriber Bulk Import Status

The Subscriber Bulk Import Status page is a monitoring page, which is read-only. This page allows the operator to monitor the Subscriber Bulk Import status, which initiates on running the Subscriber Bulk Import tool. For more information, see [bulk import tool](#).

To access the Subscriber Bulk Import Status page, click **UDR** and then, **Subscriber Bulk Import Status** in the left navigation pane of the CNC Console application. The following page appears:

Figure 3-5 Subscriber Bulk Import Status Page

Import File	Time Queued	Time Started	Time Completed	Progress (%)	Pass Count	Fail Count	Total Count	Status	Failed Keys
No data to display.									

The Subscriber Bulk Import Status page shows following bulk import transaction details:

- **Import File:** shows the csv file name that has bulk import data.
- **Time Queued:** shows the time when bulk import transaction was initiated.
- **Time Started:** shows the time when bulk import transaction begins.
- **Time Completed:** shows the time when bulk import transaction completes.
- **Progress (%):** shows the percentage completion of bulk import.
- **Pass Count:** shows the number of records that have been successfully imported.
- **Fail Count:** shows the number of records that failed to import.
- **Total Count:** shows the total number of records in the import file.
- **Status:** shows the overall status of the bulk import transaction.
- **Failed Keys:** shows information about those transactions, which are not provisioned (by bulk import tool) due to any failure. CNC Console displays failed keys only if the failed key count is less than 100.

Note

Using **Filter** option, you can filter the records you are interested in and using **Refresh** option, you can refresh the Subscriber Bulk Import Status page.

3.7 Configuration Export & Import

This section describes how to perform bulk export and bulk import of the UDR configuration data using CNC Console.

In the CNC Console application, click **UDR** and then **Export & Import**. The **Configuration Export & Import** page appears.

Figure 3-6 Configuration Export & Import

Configuration Export & Import					
Export			Import		
			Refresh Bulk Export		
Export ResourceId	Export TimeStamp	Progress(%)	Status	Actions	Result
7e67b1e0-af23-4adf-aff-29b66f1a61bb	Thu, 05 Aug 2021 05:06 UTC	100	DONE	Download Report	PARTIAL_SUCCESS

The Configuration Export & Import page has two tabs: Export (default) and Import. This page allows you to:

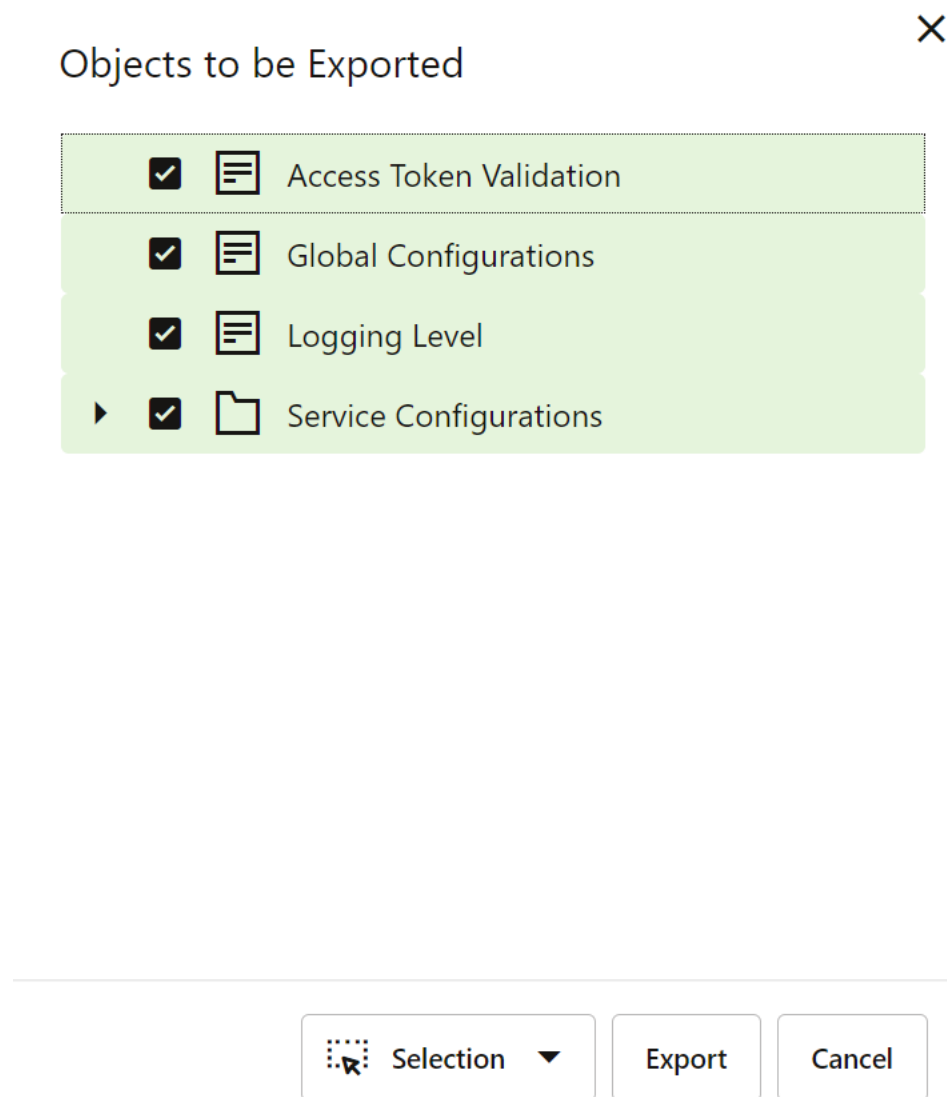
- Export the configuration data.
- Download the exported data.
- Import the data (zip file) that you have exported earlier.
- Generate report to view details of successful and failed configurations.

Exporting Configuration Data

To export the configuration data:

1. Click the **Bulk Export** button in the Export tab. The **Objects to be Exported** dialog box appears.

Figure 3-7 Objects to be Exported Dialog Box



2. Select the objects that you want to export. Click the **Selection** button to either 'Select All' or 'Deselect All' the objects.
3. Click **Export** to export the selected objects data. The data in the Configuration Export & Import page is updated.
4. Click **Cancel** to exit the dialog box.

Downloading Exported Configuration Data

To download the exported configuration data:

1. Click the **Export Configuration Download** icon available under the **Actions** column of the **Export** tab.
2. The "Export Data downloaded Successfully" message appears and the zip file of the exported data is saved to your system. For example:
exportedFiles_1.12.41_20210807133529.zip

Downloading Exported Configuration Data Report

To download the exported configuration data report:

1. Click the **Export Report Download** icon available under the **Actions** column of the **Export** tab.
2. The "Export status report downloaded successfully" message appears and the export report in the .txt file format is saved to your system. For example:
exportReport_1.12.41_20210807134056.txt

Importing Configuration Data

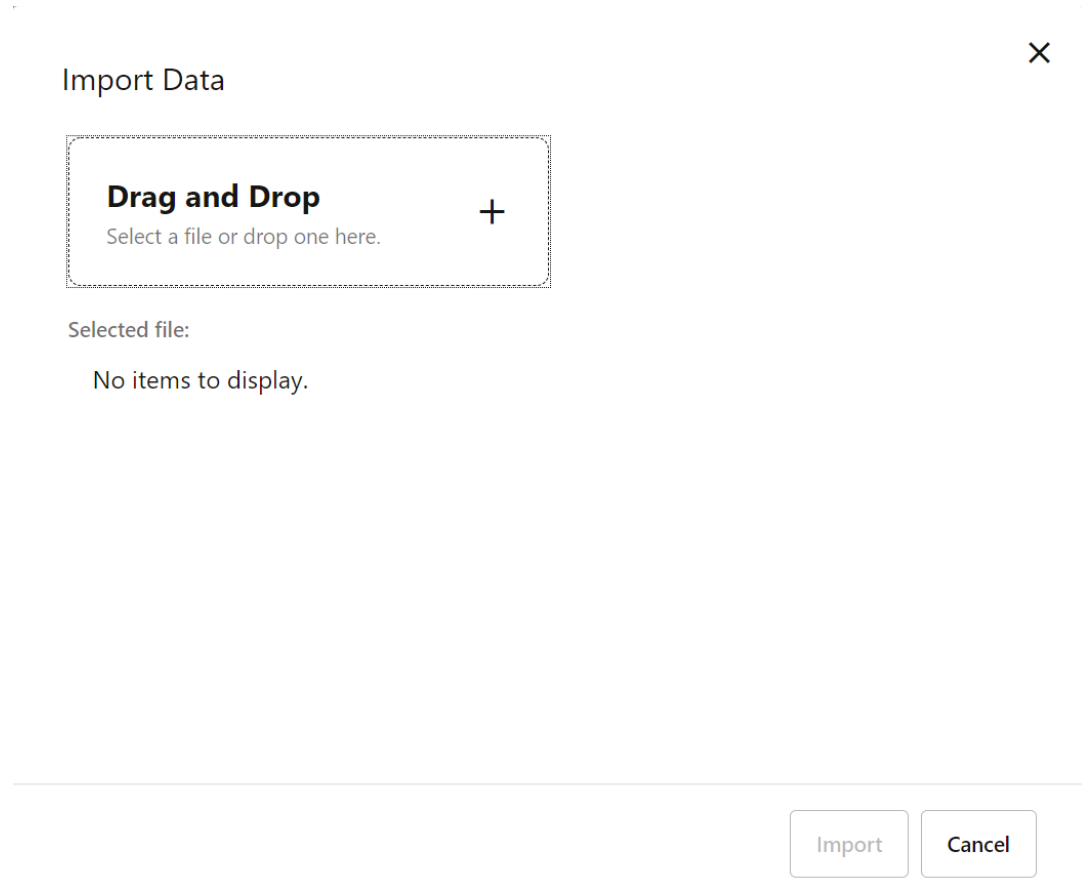
To import the configuration data:

1. Navigate to the **Import** tab. The **Configuration Export & Import** page appears as follows:

Figure 3-8 Importing Configuration Data

Configuration Export & Import					
Export			Import		
			Refresh Import		
Import ResourceId	Import TimeStamp	Progress(%)	Status	Actions	Result
6c456c8c-dfbc-4ada-8183-242eefc6c2c3	Thu, 05 Aug 2021 05:06 UTC	100	DONE		PARTIAL_SUCCESS

2. Click the **Import** button. The **Import Data** dialog box appears.

Figure 3-9 Import Data Dialog Box

3. Select the file that has configuration data and click **Import**.

Note

You can only import the configuration data that you had exported earlier.

4. The data in the **Configuration Export & Import** page is updated.
5. Click **Cancel** to exit the dialog box.

Downloading Imported Configuration Data Report

To download the imported configuration data report:

1. Click the **Import Report Download** icon available under **Actions** column of the **Import** tab.
2. The "Import status report downloaded successfully" message appears and the import report in the .txt file format is saved to your system. For example:
importReport_1.12.41_20210807134056.txt

Using REST APIs for Export & Import

UDR provides REST APIs to bulk export and bulk import its configuration data. For more information about REST APIs, see the **Configuration APIs for Import and Export** section of the *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*.

3.8 Subscriber Export Tool Status

The Subscriber Export Tool Status page is a monitoring page, which is read-only. This page allows the operator to monitor the Subscriber Export Tool Status, which initiates on running the Subscriber Export tool. For more information, see [Subscriber Export Tool](#).

To access the Subscriber Export Tool Status page, click **UDR** and then, **Subscriber Export Tool Status** in the left navigation pane of the CNC Console application. The following page appears:

Figure 3-10 Subscriber Export Tool Status

Job Id	Subscriber Count	Exported Subscriber Count	Job Progress	Status	Export State
No data to display.					

The Subscriber Export Tool Status page shows following export tool transaction details:

- **Job Id:** Generates unique ID for export.
- **Subscriber Count:** Total subscriber count present in the database
- **Exported Subscriber Count:** Total subscriber count exported from the database.
- **Job Progress:** Shows the progress of dump in percentage.
- **Status:** Shows the overall status of the export transaction. For example, RUNNING, COMPLETED, SKIPPED, or FAILED.
- **Export State:** This field indicates number of completed stages. For example, export from database is successful (1/1) and transfer of data from path to remote is successful (2/2).

Note

Using **Filter** option, you can filter the records you are interested in and using **Refresh** option, you can refresh the Subscriber Export Tool Status page.

3.9 Global Configurations

To access the **Global Configurations** page, click UDR and then, Global Configurations in the left navigation pane of the CNC Console application. The following page appears:

Figure 3-11 Global Configurations

Global Configurations Edit Refresh

DB Service Name:

DB Service Name
mysql-connectivity-service.occne-rs1

UDR Services:

UDR Services
n5g-eir-eic

UDSF Enabled:

UDSF Enabled
false

Ingress HTTP Port:

Ingress HTTP Port

Ingress HTTPS Port:

Ingress HTTPS Port

SNSSAI:

SNSSAI
2-FFFFFF

DNN:

DNN
dnn1

Consumer NF:

Consumer NF
PCF,UDM,NEF

Auto Create Subscriber:

Auto Create Subscriber
true

SBI Correlation Info Header Enable:

SBI Correlation Info Header Enable
false

NF InstanceId:

NF InstanceId
5a7bd676-ceeb-44bb-95e0-f6a55a328b03

VSA Default Billing Day:

VSA Default Billing Day
1

Subscriber Activity Enabled:

Subscriber Activity Enabled
false

Add Default Billing Day:

Add Default Billing Day
true

Auto Enroll On Signalling:

Auto Enroll On Signalling
true

E-tag Enabled:

E-tag Enabled
true

Enable Controlled Shutdown:

Enable Controlled Shutdown
false

OnDemandMigration Key Type:

OnDemandMigration Key Type
msisdn

OnDemandMigration Key Range:

OnDemandMigration Key Range
000000-000000

Enable Suppress Notification:

Enable Suppress Notification
true

Subscriber Identifiers

NAI:

NAI

IMSI:

IMSI

EXTID:

EXTID

MSISDN:

MSISDN

Access Log Enabled:

Access Log Enabled
true

Default EIR Response:

Default EIR Response
EQUIPMENT_UNKNOWN

Default EIR Response when IMEI not present:

Default EIR Response when IMEI not present
EQUIPMENT_UNKNOWN

Imei Lookup Fallback Enabled:

Imei Lookup Fallback Enabled
false

s13 Interface Enabled:

s13 Interface Enabled
false

The Global Configurations page displays the value of some global parameters as follows:

Table 3-38 Global Configurations Parameters

Field Name	Default Value	Attribute	Description
DB Service Name	mysql-connectivity-service.occne-db	Read-only	Displays the name of the DB Service.
UDR Services	All	Read-only	Displays 'All' when all the UDR services are engaged.
UDSF Enabled	false	Read-only	Displays 'true' when UDSF is enabled and 'false' when UDSF is not enabled.

Table 3-38 (Cont.) Global Configurations Parameters

Field Name	Default Value	Attribute	Description
Ingress HTTP Port	80	Read-only	Defines the Ingress Gateway signaling port for HTTP.
Ingress HTTPS Port	443	Read-only	Defines the Ingress Gateway signaling port for HTTPS.
SNSSAI	2-FFFFFF	Read-only	Single – Network Slice Selection Assistance Information (SNSSAI) is used to uniquely identify a Network Slice. The SNSSAI contains Slice/Service Type (SST) and an optional Slice Differentiators (SD) components.
DNN	dnn1	Read-only	Used to identify 5G Data Networks. The DNN is typically in the form of an Access Point Name (APN).
Consumer NF	PCF,UDM,NEF	Read-only	Displays the Consumer network functions for UDR
Auto Create Subscriber	true	Editable	Indicates the flag to enable or disable the auto creation of subscriber when the PUT operation is performed on a new UEID.
SBI Correlation Info Header Enable	false	Editable	If the flag is enabled, the 3gpp-Sbi-Correlation-Info header is added in the dr-services responses and notifications requests in Egress GW.
NF InstanceId	5a7bd676- ceeb-44bb-95e0- f6a55a328b03	Read-only	Indicates the NF Instance ID for UDR. The same is registered with NRF.
VSA Default Billing Day	1	Read-only	Indicates the billing day value.
Subscriber Activity Enabled	false	Editable	Enables or disables the subscriber activity
Add Default Billing Day	true	Editable	Indicates the flag to enable or disable the add default billing day.

Table 3-38 (Cont.) Global Configurations Parameters

Field Name	Default Value	Attribute	Description
Auto Enroll On Signalling	true	Editable	Indicates the flag to enable or disable the auto enroll of subscriber data. Note: This flag is applicable only for sm-data. It is not applicable for am-data and UE policyset.
E-tag Enabled	false	Editable	Indicates the flag to enable or disable the ETag.
Enable Controlled Shutdown	false	Editable	Indicates the flag to enable or disable the operational state.
OnDemandMigration Key Type	"msisdn"	Editable	Indicates the type of the key for the keyRange for On-demand. The keyType can either be "msisdn" or "imsi".
OnDemandMigration Key Range	"000000-000000"	Editable	This defines the range of subscribers for On-demand. Multiple key ranges are not allowed.
Enable Suppress Notification	true	Editable	Indicates the flag to enable or disable the suppress notification.
Subscriber Identifiers	NA	NA	Configure different subscriber identifiers for each keys.
NAI	NA	Editable	Subscriber identifier can be configure up to 100 keys.
IMSI	NA	Editable	Subscriber identifier can be configure up to 100 keys.
EXTID	NA	Editable	Subscriber identifier can be configure up to 100 keys.
MSISDN	NA	Editable	Subscriber identifier can be configure up to 100 keys.
Access Log Enabled	true	Editable	Flag to enable or disable the Access Log feature.

Table 3-38 (Cont.) Global Configurations Parameters

Field Name	Default Value	Attribute	Description
Default EIR Response	EQUIPMENT_UNKNOWN	Editable	Flag to set the default EIR response. Accepted values are as follows: • EQUIPMENT_UNKNOWN • WHITELISTED • GREYLISTED • BLACKLISTED
Default EIR Response when IMEI not present	EQUIPMENT_UNKNOWN	Editable	Flag to set the default EIR response when IMEI is not found in the EIR database. Accepted values are as follows: • EQUIPMENT_UNKNOWN • WHITELISTED • GREYLISTED • BLACKLISTED
Imsi Lookup Fallback Enabled	true	Editable	Flag to enable or disable the IMSI lookup fallback feature.
s13 Interface Enabled	true	Editable	Flag to enable or disable diameter S13 interface feature.

Note

These values are extracted from the database and on every Helm upgrade, users need to click the **Refresh** icon to see the latest values.

3.10 Logging Level Configuration

Perform the following procedure to configure the Logging Level:

The **Logging Level Configuration** page allows its operators to view and edit the log level details at the application log level and package log level for each service.

To access the **Logging Level Configuration** page, click **UDR** and then **Logging Level** in the left navigation pane of the CNC Console application.

The **Logging Level Configuration** page appears with following log level details:

- **Service:** This field shows UDR microservice name.
- **Application Log Level:** This field shows the log level of the microservice
- **Additional Error Logging:** This flag enables or disables the error logging feature when set to enable or disable respectively.

- **Logging Subscriber Info:** This flag enables or disables the logging of UE Identifier and subscriberid. when set to enable or disable respectively.

 Note

Additional Error Logging and **Log Subscriber Info** fields are applicable for UDR microservices only.

- **View:** Click **View** corresponding to each microservice. The **View Log Level List** page appears.

Figure 3-12 Sample: View Log Level List

✕

View Log Level List

Service:

Service
ingress-gateway-prov

Application Log Level:

Application Log Level
WARN

Package Log Level:

Package Name ↕	Package Log Level ↕
root	WARN
oauth	WARN

▼ Log Discarding

Enabled:

Enabled
false

Feature To Threshold Mapping:

Feature Name ▼	Threshold Factor ↕	
No data to display.		

Editing Log Level Details

To edit log level details:

1. Click **Edit** on the Logging Level Configuration page. The **Edit Log Level** page appears.
2. Select the **Service Type** from the drop-down list to select anyone of the following NF Service type:

- **Data Repository Service**
 - **Alternate Route**
 - **Bulk Import Tool**
 - **Bulk Export Tool**
 - **Diameter Service**
 - **Diameter Gateway**
 - **Data Repository Provisioning Service**
 - **Egress Gateway**
 - **Ingress Gateway Signaling**
 - **Ingress Gateway Provisioning**
 - **Notify Service**
 - **NRF Client NF Management**
 - **On Demand Migration**
 - **App Info**
 - **Perf Info**
 - **Nudr Config Service**
3. Select the **Application Log Level** from the dropdown menu to set the constraints of the log. Possible values are:
 - **DEBUG**
 - **ERROR**
 - **INFO**
 - **TRACE**
 - **WARN**
 4. Select the **Additional Error Logging** status from the drop-down list.
 5. Select the **Log Subscriber Info** status from the drop-down list.
 6. To add the **Package Log Level** details:
 - a. Click the **Add** button available in the **Package Log Level** section. The **Add Package Log Level** dialog box appears:

Figure 3-13 Add Package Log Level

Add Package Log Level ✕

Package: Required

Log Level: Log Level
DEBUG ▼

Save Cancel

- b. Enter the **Package** name and select the **Log Level** from the list. The Log Level options are:
 - DEBUG
 - ERROR
 - INFO
 - TRACE
 - WARN
- c. Click **Save** to add the package log level details. [OR] Click **Cancel** to exit the Add Package Log Level dialog box.
7. Click **Save**. A confirmation message, "Save successfully." appears.
8. If you do not want to modify any log level detail, click **Cancel**. The **Logging Level Configuration** page appears.
9. Click **Refresh** to reload the **Logging Level Configuration** page.

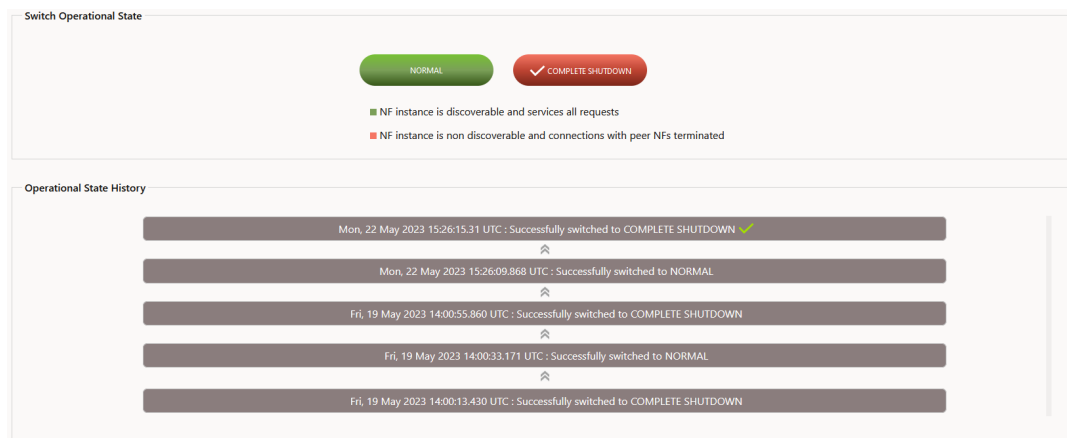
3.11 Controlled Shutdown Configurations

This section describes how to perform Controlled Shutdown configurations.

3.11.1 Operational State

Perform the following steps to change the Operational State of a site:

- In the CNC Console application, under **UDR**, click **Controlled Shutdown**, and then select **Operational State**. This opens the page displaying the two groups, **Switch Operational State** and **Operational State History**:

Figure 3-14 Operational State

- Switch Operational State displays the following operational states:
 - * NORMAL: NF instance is discoverable and services all requests.
 - * COMPLETE SHUTDOWN: NF instance is non discoverable and no new session creation requests accepted.

Note

The current state of any site can be identified with a tick mark.

You can switch to a different operational state by clicking the NORMAL or COMPLETE SHUTDOWN button.

- Operational State History: It displays the history of the operational states along with the timestamp.

3.12 Provisioning

To configure UDR services using CNC Console, you can use Provisioning. In the left navigation pane of the CNC Console application, click **UDR** and then **Provisioning**.

Using **Provisioning**, UDR users can configure following services:

- [Provisioning - NEF Data](#)
- [Provisioning - Profile Data](#)
- [Provisioning - EIR Data](#)
- [Provisioning - PCF Data](#)
- [Provisioning - SLF Data](#)
- [Provisioning - UDM Data](#)
- [Provisioning - Schema Management Data](#)

Using **Provisioning**, SLF users can configure following services:

- [Provisioning - Profile Data](#)

- [Provisioning - SLF Data](#)
- [Provisioning - Schema Management Data](#)

Using **Provisioning**, EIR users can configure following services:

- [Provisioning - Profile Data](#)
- [Provisioning - EIR Data](#)
- [Provisioning - Schema Management Data](#)

3.12.1 Provisioning - NEF Data

This section describes how to provision operations for NEF Data.

In the CNC Console application, click **UDR** and then **Provisioning**. From the **Provisioning** menu, select **NEF Data** in the left navigation pane. The **Provisioning Operations for NEF Data** page appears:

Figure 3-15 Provisioning Operations for NEF Data

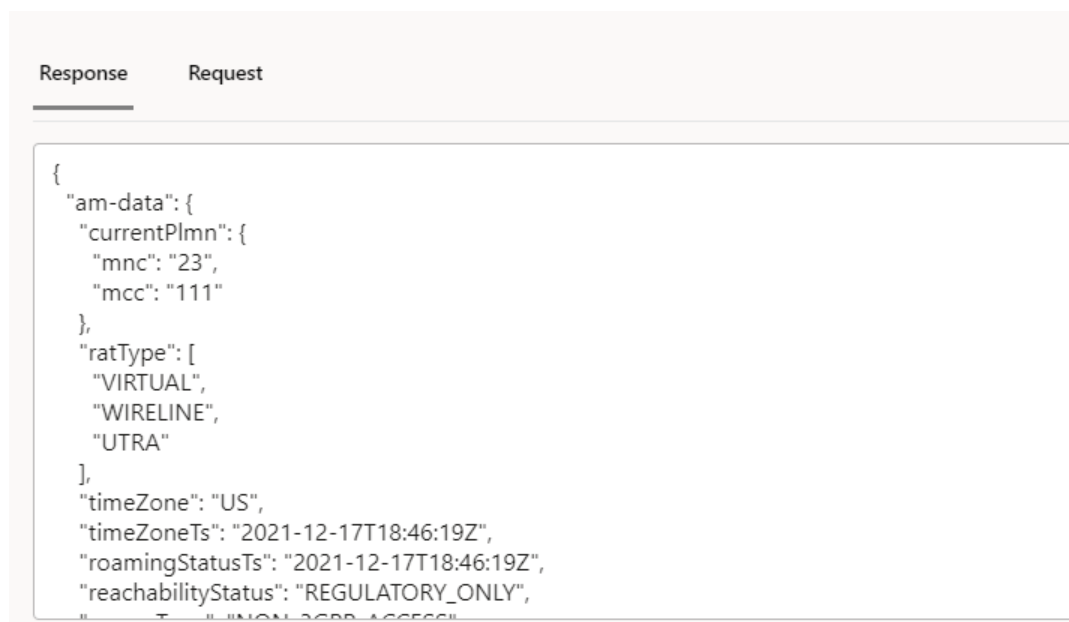
In the Provisioning Operations for NEF Data page, users can:

- Extract NEF data from UDR database based on the UE ID entered
- Add new NEF data to the UDR database
- Modify an existing NEF data on the based on a UE ID
- Delete existing NEF data that is not needed in the UDR database

Extracting Profile Data

To extract profile data:

1. Ensure **Resource Name** is set to 'NEF Data'
2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Get**. A confirmation message, '**Fetchd Successfully**' appears for a second, and the result appears in the **Response** text area as follows:

Figure 3-16 Sample Page: Get - NEF Data - Response

4. An **Error message summary** appears in case of incorrect UE ID entered.

Adding NEF Data

To add NEF data:

1. Ensure **Resource Name** is set to 'NEF Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Add**. The **Request** text area appears.
4. Enter the NEF data that you want to add to the UDR database.
5. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If you enter an incorrect input, an Error message summary appears.

Editing Profile Data

To edit profile data:

1. Ensure **Resource Name** is set to 'NEF Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Edit**. The **Request** text area displays existing NEF data.
4. Enter the updated NEF data.
5. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If you enter an incorrect input, an Error message summary appears.

Deleting NEF Data

To delete NEF data:

1. Ensure **Resource Name** is set to 'NEF Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Delete**.
4. The '**Deleted Successfully**' message appears.

Note

If you enter an incorrect input, an Error message summary appears.

Clearing NEF Data

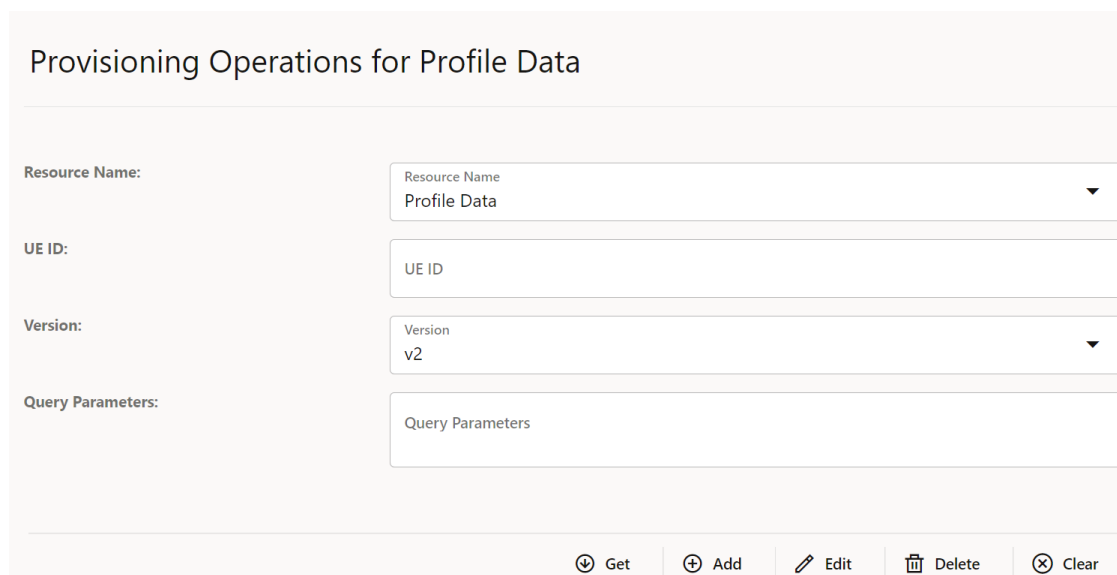
Click the **Clear** option to clear all the text fields and text areas of the Provisioning Operations for NEF Data.

3.12.2 Provisioning - Profile Data

This section describes how to operate provisioning operations for Profile Data.

In the CNC Console application, click **UDR** and then **Provisioning**. From the **Provisioning** menu, select **Profile Data** in the left navigation pane. The **Provisioning Operations for Profile Data** page appears:

Figure 3-17 Provisioning Operations for Profile Data



Provisioning Operations for Profile Data

Resource Name:

UE ID:

Version:

Query Parameters:

 Get  Add  Edit  Delete  Clear

In the Provisioning Operations for Profile Data page, user can:

- Extract profile data from UDR database on the basis of UE ID entered
- Add new profile data to the UDR database

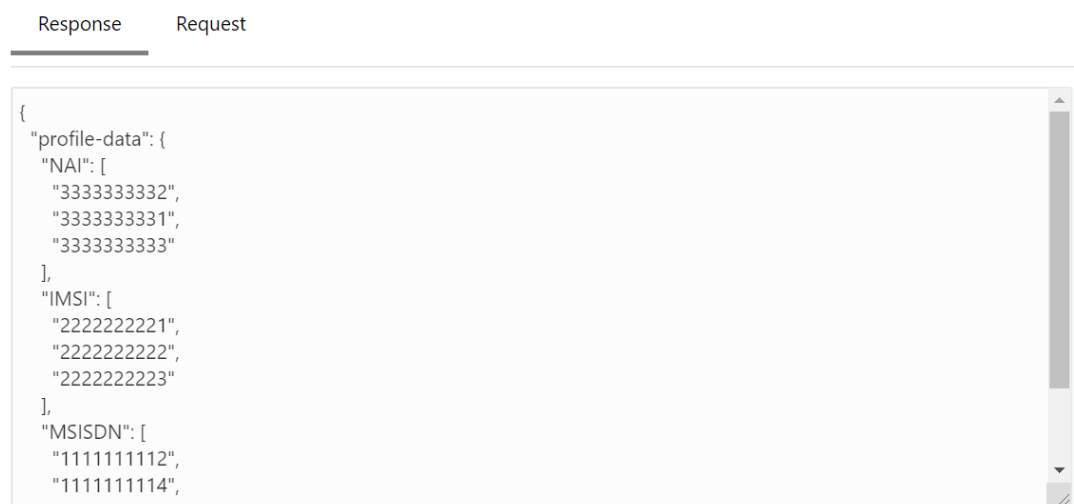
- Modify an existing profile data on the basis of UE ID
- Delete an existing profile data that is no more needed in the UDR database

Extracting Profile Data

To extract a profile data:

1. Ensure **Resource Name** is set to 'Profile Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Get**. A confirmation message, '**Fetched Successfully**' appears for a second and the result appears in the **Response** text area as follows:

Figure 3-18 Sample Page: Get - Profile Data - Response



4. An **Error message summary** appears in case of incorrect UE ID entered.

Adding a Profile Data

To add a profile data:

1. Ensure **Resource Name** is set to 'Profile Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Add**. The **Request** text area appears.
4. Type-in the profile data that you want to add to the UDR database.
5. Click **Submit**. The '**Saved Successfully**.' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Editing a Profile Data

To edit a profile data:

1. Ensure **Resource Name** is set to 'Profile Data'.

2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Edit**. The **Request** text area displays an existing profile data.
4. Type-in the updated profile data.
5. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Deleting a Profile Data

To delete a profile data:

1. Ensure **Resource Name** is set to 'Profile Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111114
3. Click **Delete**.
4. The '**Deleted Successfully**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Clearing a Profile Data

Click the **Clear** option to clear all the text fields and text areas of the Provisioning Operations for Profile Data.

3.12.3 Provisioning - EIR Data

This section describes provisioning operations for EIR Data.

In the CNC Console application, click **UDR** and then **Provisioning**. From the **Provisioning** menu, select **EIR Data** in the left navigation pane. The **Provisioning Operations for EIR Data** page appears:

Figure 3-19 Provisioning GUI for EIR Data

Provisioning Operations for EIR Data

Resource Name: Resource Name
EIR Data

UE ID: UE ID

Version: Version
v1

Get Add Edit Delete Clear

In the Provisioning Operations for EIR Data page, users can:

- Extract EIR data from UDR database based on the UE ID entered
- Add new EIR data to the UDR database
- Modify an existing EIR data on the based on a UE ID
- Delete existing EIR data that is not needed in the UDR database

Extracting Profile Data

To extract profile data:

1. Ensure **Resource Name** is set to 'EIR Data'
2. Type an appropriate **UE ID**. For example: UE ID - imei-1111111111
3. Click **Get**. A confirmation message, '**Fetches Successfully**' appears for a second, and the result appears in the **Response** text area as follows:

Figure 3-20 Sample Page: Get - EIR Data - Response

Response	Request
<pre>{ "profile-data": { "MSISDN": ["1111111112", "1111111113"], "IMSI": ["1111111121", "1111111131"], "IMEI": ["1111111111"] }, "equipment-status": { "white": true, "black": true, "grey": true } }</pre>	

4. An **Error message summary** appears in case of incorrect UE ID entered.

Adding EIR Data

To add EIR data:

1. Ensure **Resource Name** is set to 'EIR Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111112
3. Click **Add**. The **Request** text area appears.
4. Enter the EIR data that you want to add to the UDR database.
5. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If you enter an incorrect input, an Error message summary appears.

Editing Profile Data

To edit profile data:

1. Ensure **Resource Name** is set to 'EIR Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111112
3. Click **Edit**. The **Request** text area displays existing NEF data.
4. Enter the updated EIR data.
5. Click **Submit**. The '**Saved Successfully**.' message appears.

Note

If you enter an incorrect input, an Error message summary appears.

Deleting EIR Data

To delete EIR data:

1. Ensure **Resource Name** is set to 'EIR Data'.
2. Type an appropriate **UE ID**. For example: msisdn-1111111112
3. Click **Delete**.
4. The '**Deleted Successfully**' message appears.

Note

If you enter an incorrect input, an Error message summary appears.

Clearing EIR Data

Click the **Clear** option to clear all the text fields and text areas of the Provisioning Operations for EIR Data.

3.12.4 Provisioning - PCF Data

This section describes how to operate provisioning operations for PCF (Policy Control Function) Data.

In the CNC Console application, click **UDR** and then **Provisioning**. From the **Provisioning** menu, select **PCF Data** in the left navigation pane. The **Provisioning Operations for PCF Data** page appears:

Figure 3-21 Provisioning Operations for PCF Data

Provisioning Operations for PCF Data

Resource Name: Policy Data

UE ID:

Version: v2

Query Parameters:

In the Provisioning Operations for PCF Data page, a user can:

- Select the **Resource Name** from the drop-down list.
- Extract PCF data from UDR database on the basis of UE ID entered.
- Add PCF Data on the basis of payload.
- Modify the existing PCF data on the basis of UE ID.
- Delete the existing PCF data that is no more needed in the UDR database.

Extracting PCF Data

To extract the PCF data:

1. Select the **Resource Name** from the drop-down list.
2. Type an appropriate **UE ID**. For example: msisdn-9111111112.
3. Click **Get**. A confirmation message, '**Fetchd Successfully.**' appears and the result appears in the **Response** text area as follows:

Figure 3-22 Provisioning PCF Get Response

Response	Request
<pre>{ "pralnfos": { "p1": { "prald": "p1", "ecgiList": [{ "plmnlId": { "mcc": "976", "mnc": "32" }, "eutraCellId": "92FFdBE" }, { "plmnlId": { "mcc": "977", "mnc": "32" }, "eutraCellId": "92FFdBE" }] } } }</pre>	

4. An **Error message summary** appears in case of incorrect UE ID entered.

Adding a PCF Data

To add a PCF data:

1. Select the **Resource Name** from the drop-down list.
2. Type an appropriate **UE ID**. For example: msisdn-9111111112.
3. Click **Add**. The **Request** text area displays an existing PCF data.
4. Type-in the updated profile data.
5. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Editing a PCF Data

To edit a PCF data:

1. Select the **Resource Name** from the drop-down list.
2. Type an appropriate **UE ID**. For example: msisdn-9111111112.
3. Click **Edit**. The **Request** text area displays an existing PCF data.
4. Type-in the updated profile data.
5. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Deleting a PCF Data

To delete a PCF Data:

1. Select the **Resource Name**.
2. Type an appropriate **UE ID**. For example: msisdn-9111111112.
3. Click **Delete**.
4. The '**Deleted Successfully**.' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Clearing a PCF Data

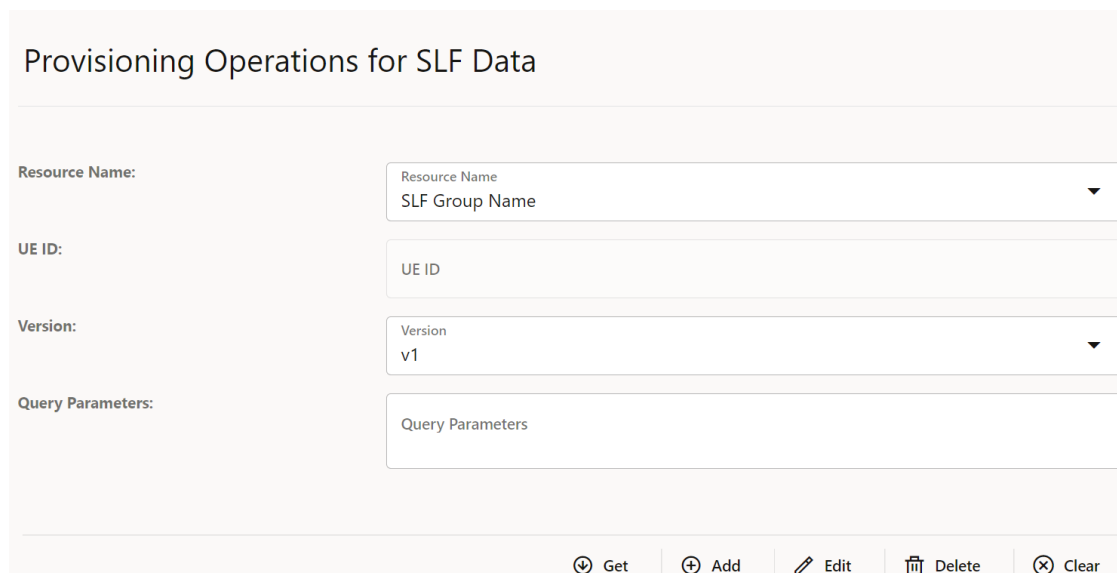
Click the **Clear** option to clear the text fields and text areas of the Provisioning Operations for PCF Data page.

3.12.5 Provisioning - SLF Data

In this section, you will learn to operate provisioning operations for SLF (Subscriber Location Function) Data.

In the CNC Console application, click **UDR** and then **Provisioning**. From the **Provisioning** menu, select **SLF Data** in the left navigation pane. The **Provisioning Operations for SLF Data** page appears:

Figure 3-23 Provisioning Operations for SLF Data



Provisioning Operations for SLF Data

Resource Name: SLF Group Name

UE ID:

Version: v1

Query Parameters:

 Get  Add  Edit  Delete  Clear

Note

SLF Group Name must be provisioned before SLF Subscriber is provisioned.

In the Provisioning Operations for SLF Data page, user can:

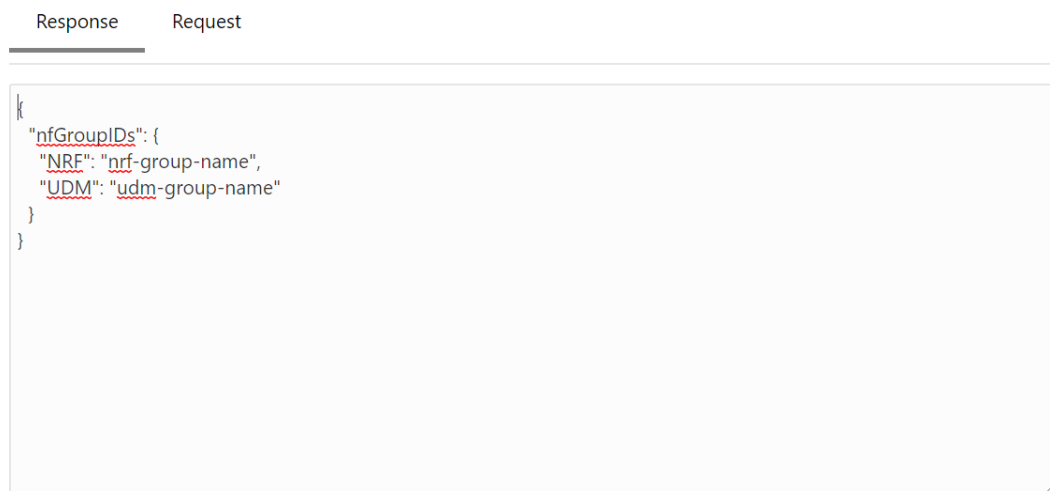
- Extract SLF data from UDR database on the basis SLF Group Name (Query Parameter) or SLF Subscriber (UEID).
- Add or edit an existing SLF data on basis of SLF Group Name (Query Parameter) or SLF Subscriber (UEID).
- Delete an existing SLF data that is no more needed in the UDR database.

Extracting SLF Data on the basis of SLF Group Name

To extract the SLF data on the basis of SLF Group Name:

1. Select the **Resource Name** as SLF Group Name.
2. Enter the **Query Parameters**.
3. Click **Get**. A confirmation message, '**Fetchd Successfully.**' appears and the result appears in the **Response** text area as follows:

Figure 3-24 Provisioning SLF Get Response



Extracting SLF Data on the basis of SLF Subscriber

To extract the SLF data on the basis of SLF Subscriber:

1. Select the **Resource Name** as SLF Subscriber.
2. Type an appropriate **UE ID**. For example: msisdn-1111111112
3. Click **Get**. A confirmation message, '**Fetchd Successfully.**' appears and the result appears in the **Response** text area as follows:

Figure 3-25 Provisioning SLF Get Response

Response	Request
<pre> { "nfGroupIDs": { "NRF": "nrf-group-name", "UDM": "udm-group-name" } } </pre>	

4. An **Error message summary** appears in case you enter an incorrect UE ID.

Adding SLF Data on the basis of SLF Group Name or SLF Subscriber

To add the SLF data:

1. Select the **Resource Name** as either SLF Group Name or SLF Subscriber.
2. If you have selected the Resource Name as SLF Group Name then,
 - a. Enter the **Query Parameters**.
 - b. Click **Add**. The **Request** text area appears.
 - c. Type the SLF data.
 - d. Click **Submit**. The '**Saved Successfully.**' message appears.

[OR]

3. If you have selected the Resource Name as SLF Subscriber then,
 - a. Type an appropriate **UE ID**. For example: msisdn-111111112.
 - b. Click **Add**. The **Request** text area appears.
 - c. Type-in the SLF data.
 - d. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Editing SLF Data

To edit a SLF data:

1. Select the **Resource Name** as either SLF Group Name or SLF Subscriber.
2. If you have selected the Resource Name as SLF Group Name then,

- a. Type in the **Query Parameters** and click **Get**. The result appears in the **Response** text area.
- b. Click **Edit**. The **Request** text area displays an existing SLF data. Modify the relevant details.
- c. Click **Submit**. The '**Saved Successfully.**' message appears.

[OR]

3. If you have selected the Resource Name as SLF Subscriber then,
 - a. Type in the **UE ID** and click **Get**. The result appears in the **Response** text area.
 - b. Click **Edit** and modify the required details. The **Request** text area displays an existing SLF data. Modify the relevant details.
 - c. Click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Deleting a SLF Data

To delete a SLF Data:

1. Select the **Resource Name** as either SLF Group Name or SLF Subscriber.
2. If you have selected the Resource Name as SLF Group Name then enter the 'Query Parameters' and if you have selected the Resource Name as SLF Subscriber then enter the 'UE ID'. **For example:** msisdn-111111112
3. Click **Delete**.
4. The '**Deleted Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Clearing a SLF Data

Click the **Clear** option to clear the text fields and text areas of the Provisioning Operations for SLF Data page.

3.12.6 Provisioning - UDM Data

This section describes how to operate provisioning operations for UDM (Unified Data Management) Data.

In the CNC Console application, click **UDR** and then **Provisioning**. From the **Provisioning** menu, select **UDM Data** in the left navigation pane. The **Provisioning Operations for UDM Data** page appears:

Figure 3-26 Provisioning Operations for UDM Data

Provisioning Operations for UDM Data

Resource Name:

Resource Name
Access And Mobility Subscription Data

UE ID:

UE ID

PDU Session ID:

PDU Session ID

Serving PLMN ID:

Serving PLMN ID

Version:

Version
v1

Query Parameters:

Query Parameters

⬇ Get

⬆ Add

✎ Edit

🗑 Delete

⊗ Clear

In the Provisioning Operations for UDM Data page, user can extract, modify and delete UDM data on the basis of Resource Name, UE ID, PDU Session ID, Serving PLMN ID, Version, and Query Parameters entered.

Extracting UDM Data

To extract the UDM data:

1. Select the **Resource Name** from the drop-down list.
2. Type an appropriate **UE ID**, **PDU Session ID** and **Serving PLMN ID** in their respective fields.
3. Select an appropriate **Version** and enter **Query Parameters**.
4. Click **Get**. A confirmation message, '**Fetchd Successfully.**' appears and the result appears in the **Response** text area.
5. An **Error message summary** appears in case of incorrect UE ID entered.

Adding a UDM Data

To add a UDM data:

1. Select the **Resource Name**.
2. Type an appropriate **UE ID**, **PDU Session ID** and **Serving PLMN ID** in their respective fields.
3. Select an appropriate **Version** and enter **Query Parameters**.
4. Click **Edit**. The **Request** text area appears.

5. Enter details and click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Editing a UDM Data

To edit a UDM data:

1. Select the **Resource Name**.
2. Type an appropriate **UE ID**, **PDU Session ID**, and **Serving PLMN ID** in their respective fields.
3. Select an appropriate **Version** and enter **Query Parameters**.
4. Click **Edit**. The **Request** text area displays an existing UDM data.
5. Edit as required and click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Deleting a UDM Data

To delete a UDM Data:

1. Enter the **Resource Name**.
2. Type an appropriate **UE ID**, **PDU Session ID**, and **Serving PLMN ID** in their respective fields.
3. Select an appropriate **Version** and enter **Query Parameters**.
4. Click **Delete**.
5. The '**Deleted Successfully.**' message appears.

Note

If a user enters any incorrect input, an Error message summary appears.

Clearing a UDM Data

Click the **Clear** option to clear the text fields and text areas of the Provisioning Operations for UDM Data page.

3.12.7 Provisioning - Schema Management Data

This section describes how to operate provisioning operations for schema.

In the CNC Console application, click **UDR** and then **Provisioning**. From the **Provisioning** menu, select **Schema Management Data** in the left navigation pane. The **Schema Management** page appears:

Figure 3-27 Schema Management

Schema Management

Resource Name:

Resource Name
Profile Data

API Version:

API Version
v1

Query Parameters:

Query Parameters

Get

Edit

Delete

Rollback

Clear

In the **Schema Management** page, user can extract, modify, and delete a schema on the basis of **Resource Name** and **Query Parameters**.

Extracting a Schema

To extract a schema:

1. Select the **Resource Name**.
2. Enter the **Query Parameters**.
3. Click **Get**. A confirmation message, '**Fetchd Successfully.**' appears and the result appears in the **Response** text area as follows:

Figure 3-28 Schema - Get - PCF AM Data

Resource Name: PCF AM Data

Get

Edit

Delete

Clear

Response

Request

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "subscCats": {
      "minItems": "1",
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "pralnfos": {
      "patternProperties": {
        "^(.*)$": {
          "type": "object",

```

4. An **Error message summary** appears in case of incorrect UE ID entered.

Adding or Editing a Schema

To add or edit a schema:

1. Select the **Resource Name**.
2. Enter the **Query Parameters**.
3. Click **Edit**. The **Request** text area displays the selected schema.
4. Edit as required and click **Submit**. The '**Saved Successfully.**' message appears.

Note

If a user edits the schema inappropriately, an Error message summary appears.

Deleting a Schema

To delete a schema:

1. Select the **Resource Name**.
2. Enter the **Query Parameters**.
3. Click **Delete**.
4. The '**Deleted Successfully**.' message appears.

Note

If a user selects a schema that does not exist, an Error message summary appears.

Clearing a Schema Management Data

Click the **Clear** option to clear the text fields and text areas of the Schema Management page.

3.13 Service Configurations

To configure UDR services using CNC Console, you can use Service Configurations. In the left navigation pane of the CNC Console application, click **UDR** and then **Service Configurations**.

Using **Service Configurations**, UDR users can configure following services:

- [Subscriber Bulk Import Configurations](#)
- [Data Repository Service](#)
- [Diameter Gateway Service](#)
- [Diameter Gateway Congestion](#)
- [Provisioning Data Repository Service](#)
- [Notify Service](#)
- [On-Demand Migration](#)
- [Subscriber Export Tool Configurations](#)

Using **Service Configurations**, SLF users can configure following services:

- [Subscriber Bulk Import Configurations](#)
- [Data Repository Service](#)
- [Provisioning Data Repository Service](#)
- [Subscriber Export Tool Configurations](#)

Using **Service Configurations**, EIR users can perform the following configurations:

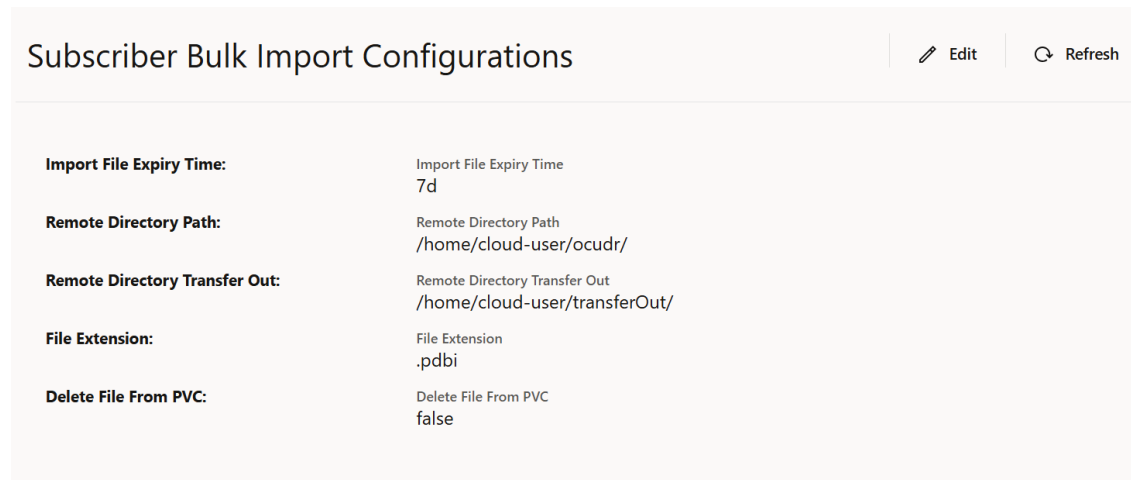
- [Subscriber Bulk Import Configurations](#)
- [Data Repository Service](#)

- [Provisioning Data Repository Service](#)
- [Subscriber Export Tool Configurations](#)

3.13.1 Subscriber Bulk Import Configurations

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Subscriber Bulk Import Configurations**. The **Subscriber Bulk Import Configurations** page appears:

Figure 3-29 Subscriber Bulk Import Configurations



Field Name	Default Value	Attribute	Description
Import File Expiry Time:	Import File Expiry Time 7d		
Remote Directory Path:	Remote Directory Path /home/cloud-user/ocudr/		
Remote Directory Transfer Out:	Remote Directory Transfer Out /home/cloud-user/transferOut/		
File Extension:	File Extension .pdbi		
Delete File From PVC:	Delete File From PVC false		

The following table explains the parameters in the **Subscriber Bulk Import Configurations** page in detail:

Table 3-39 Subscriber Bulk Import Configurations Parameters

Field Name	Default Value	Attribute	Description
Import File Expiry Time	7d	Editable	Expiry time for all the PDBI files including the result log files.
Remote Directory Path	/home/cloud-user/ocudr/	Editable	Remote host path
Remote Directory Transfer Out	/home/cloud-user/transferOut/	Editable	The log files are transferred to this remote host server path when the bulk-import processing is completed.
File Extension	.pdbi	Editable	Shows the file type
Delete File From PVC	false	Editable	Flag to enable or disable the delete file from PVC feature

Editing Subscriber Bulk Import Configurations

To update or modify configuration information related to data repository:

1. Click **Edit**. The **Edit Subscriber Bulk Import Configurations** page appears.

2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully," appears.
3. If you do not want to modify any configuration, click **Cancel**. You can navigate back to the Bulk Import Configurations page.
4. Click **Refresh** to reload the Subscriber Bulk Import Configurations page.

3.13.2 Data Repository Service

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Data Repository Service**. The **Data Repository Service Configurations** page appears:

Figure 3-30 Data Repository Service Configurations

Data Repository Service Configurations

Hikari Pool Size:	Hikari Pool Size 100
Tracing Enabled:	Tracing Enabled false
Default Slf Group Name:	Default Slf Group Name DefaultGrp
Default GroupId Enabled:	Default GroupId Enabled false
Subscription Data Subscriptions Only:	Subscription Data Subscriptions Only false
On Demand Migration Enabled:	On Demand Migration Enabled false
Allow Subscription Recreation:	Allow Subscription Recreation false

The following table explains the parameters in the **Data Repository Service Configurations** page in detail:

Table 3-40 Data Repository Service Configuration Parameters

Field Name	Default Value	Attribute	Description
Hikari Pool Size	25	Read-only	Hikari pool connection size to be created at start up.

Table 3-40 (Cont.) Data Repository Service Configuration Parameters

Field Name	Default Value	Attribute	Description
Tracing Enabled	false	Read-only	Flag to enable tracing.
Default SLF Group Name	DefaultGrp	Editable	The default group name is provisioned. If there are any changes done to this through CNC console, then the default group must 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.
Default GroupId Enabled	false	Editable	Flag to enable or disable the Default Group ID feature.
Subscription Data Subscriptions Only	false	Read-only	Flag to enable subscription data.
On Demand Migration Enabled	false	Editable	Flag to enable or disable on-demand migration.
Allow Subscription Recreation	false	Editable	Flag to enable or disable overwriting of existing subscription for the POST request.

Editing Data Repository Service Configurations

To update or modify configuration information related to data repository:

1. Click **Edit**. The **Edit Data Repository Service Configurations** page appears.
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully," appears.
3. If you do not want to modify any configuration, click **Cancel**. You can navigate back to the Data Repository Service Configurations page.
4. Click **Refresh** to reload the Data Repository Service Configurations page.

3.13.3 Diameter Gateway Service

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Diameter Gateway**. The **Diameter Gateway Configurations** page appears:

Figure 3-31 Diameter Gateway Configurations

Diameter Gateway Configurations
Edit Refresh

Realm:

Identity:

Reconnect Delay:

Response Timeout:

Connection Timeout:

Watchdog Interval:

Reconnect Limit:

On Demand Migration Enabled:

Allowed Client Nodes For Diameter SH Ingress:

Realm
oracle.com

Identity
nudr.oracle.com

Reconnect Delay
6

Response Timeout
4

Connection Timeout
3

Watchdog Interval
6

Reconnect Limit
2

On Demand Migration Enabled
false

Type	Response Only	Host	Realm	Identity	Port	
			tekelec1275.com	local1.tekelec1.com		Add Edit Delete

The following table explains the parameters in the Diameter Gateway Configurations page in detail:

Table 3-41 Diameter Gateway Configurable Parameters

Field Name	Default Value	Attribute	Description
Realm	oracle.com	Read-only	UDR Diameter Gateway server realm.
Identity	nudr.oracle.com	Read-only	UDR Diameter Gateway identity or FQDN.
Reconnect Delay	4	Editable	Waiting time for reconnecting server peer in seconds.
Response Timeout	4	Editable	Time period to get the response from server peer in seconds.
Connection Timeout	7	Editable	Time period in which both client and server should establish the connection.
Watchdog Interval	7	Editable	Interval of time to send Diameter watchdog message periodically.
Reconnect Limit	70	Editable	Number of attempts allowed to reconnect with diameter server peer if server is unreachable or down.
On Demand Migration Enabled	False	Editable	Flag to indicate whether on demand migration is enabled or disabled.

Table 3-41 (Cont.) Diameter Gateway Configurable Parameters

Field Name	Default Value	Attribute	Description
Allowed Client Nodes For Diameter SH Ingress	<pre>"allowedClientNodes": [{ "host": "10.0.7.500", "realm": "tekelec1275.com" }, { "responseOnly": true, "type": "4GUDRondemandCon nection", "port": 7867, "identity": "local1.tekelec1. com" }], { "host": "10.75.185.161", "realm": "seagull", "responseOnly": true, "type": "server", "port": 3868, "identity": "sh- server" }]</pre>	Editable	If responseOnly parameter is present and the value is "true", then mandatory fields are: • realm • identity • host • port If responseOnly parameter is not present or the value is false, then mandatory fields are: • realm • identity
Type	NA	Editable	Type of peer node
Response only	NA	Editable	Flag to initiate connection to the peer node.

Table 3-41 (Cont.) Diameter Gateway Configurable Parameters

Field Name	Default Value	Attribute	Description
Host	NA	Editable	Peer node IP address or FQDN
Identity	seagull1a.seagull1.com	Editable	Identity of Peer node
Realm	seagull1.com	Editable	Realm of peer node
Port	NA	Editable	Peer node port used by diameter gateway when initiating connection.

Editing Diameter Gateway Configurations

To update or modify any configuration information related to diameter gateway:

1. Click **Edit**. The **Edit Diameter Gateway Configurations** page appears:
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully," appears.
3. If you do not want to modify any configuration, click **Cancel**. The Diameter Gateway Configurations page appears.
4. Click **Refresh** to reload the Diameter Gateway Configurations page.

3.13.4 Diameter Gateway Congestion

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Diameter Gateway Congestion**. The **Diameter Gateway Congestion Configurations** page appears:

Figure 3-32 Diameter Gateway Congestion

Diameter Gateway Congestion Configurations

Edit
Refresh

Enabled:

Enabled
true

Default Message Priority:

Default Message Priority
3

Sample Count:

Sample Count
10

Calculation Interval:

Calculation Interval
100

Discard Policy

Congested State

Discard Priority:

Discard Priority
1

Return ResultCode:

Return ResultCode
DIAMETER_TOO_BUSY

Danger of Congestion State

Discard Priority:

Discard Priority
3

Return ResultCode:

Return ResultCode
DIAMETER_TOO_BUSY

Thresholds

Congested State

Memory:

Memory
80

CPU:

CPU
80

Queue:

Queue
50

Danger of Congestion State

Memory:

Memory
60

CPU:

CPU
60

Queue:

Queue
30

Priority Rules:

Rule Name	Message Priority	Add
Diameter-Sh-UDR	3	Edit Delete View
Diameter-Sh-PUR	2	Edit Delete View
Diameter-Sh-SNR	1	Edit Delete View
Diameter-Sh-PNR	4	Edit Delete View

The following table explains the parameters in the Diameter Gateway Congestion Configurations page in detail:

Table 3-42 Diameter Gateway Congestion Configurable Parameters

Field Name	Default Value	Attribute	Description
enabled	true	Editable	Flag to enable or disable the Diameter Gateway Pod Congestion feature
defaultMessagePriority	2	Editable	Default diameter message priority which is not specifically assigned any priority
sampleCount	10	Editable	Sample request count for deciding pod state
calculationInterval	100	Editable	Congestion state calculation every configured time (in ms)
loadSheddingRulesMap	NA { "CONGESTED": { "state": "CONGESTED", "discardPriority": 3, "ansWithResultCode": "DIAMETER_T00_BUSY" }, "DOC": { "state": "DOC", "discardPriority": 6, "ansWithResultCode": "DIAMETER_T00_BUSY" } }	Editable	Indicates the decision for the pod in a particular state
loadSheddingRulesMap. <CONGESTED/DOC>	NA	Editable	Indicates the pod congestion state.

Table 3-42 (Cont.) Diameter Gateway Congestion Configurable Parameters

Field Name	Default Value	Attribute	Description
loadSheddingRulesMap.<CONGESTED/DOC>.state	NA	Editable	Indicates the name of the state.
loadSheddingRulesMap.<CONGESTED/DOC>.discardPriority	NA	Editable	Indicates the diameter message discard priority.
loadSheddingRulesMap.<CONGESTED/DOC>.ansWithResultCode	NA	Editable	Indicates the diameter return string result code when the message get discarded.
priorityRules	NA [{ "ruleName": "SH-UDR", "messagePriority": 5, "conditions": { "requestType": "UDR", "application": "Sh" } }]	Editable	NA
Priority Rules.Rule Name	NA	Editable	Specifies the unique name of the message priority rule.
Priority Rules.Message Priority	NA	Editable	Specifies the priority assigned to the message. It can be a number from 0 to 15.
PriorityRules.Conditions.Request Type	NA	enum	Specifies the type of message for the selected application. The supported request values for the application type are as follow: • CER • SNR • UDR • PUR • PNR • UDR_MIGRATION

Table 3-42 (Cont.) Diameter Gateway Congestion Configurable Parameters

Field Name	Default Value	Attribute	Description
Priority Rules.Conditions.Application	NA	enum	Specifies the type of application. Users can select any of the following values from the drop-down list: Sh
Thresholds	NA <pre>{ "CONGESTED": { "memory": 80, "cpu": 80, "queue": 50 }, "DOC": { "memory": 60, "cpu": 60, "queue": 30 } }</pre>	Editable	NA
Thresholds.Congested State.Memory	NA	Editable	Memory percentage of pod state
Thresholds.Congested State.CPU	NA	Editable	CPU percentage of pod state
Thresholds.Congested State.Queue	NA	Editable	Queue percentage of pod state
Thresholds.Danger of Congested State.Memory	NA	Editable	Memory percentage of pod state
Thresholds.Danger of Congested State.CPU	NA	Editable	CPU percentage of pod state
Thresholds.Danger of Congested State.Queue	NA	Editable	Queue percentage of pod state
Discard Policy.Congested State.Discard Priority	NA	Editable	Specifies the discard priority for the load shedding rule. The discard priority value can be a number from 0 to 15. Any request message with equal or lower priority is rejected.

Table 3-42 (Cont.) Diameter Gateway Congestion Configurable Parameters

Field Name	Default Value	Attribute	Description
Discard Policy.Congested State.Return ResultCode	NA	enum	Specifies the result code, returned in the answer response, when request message is rejected as part of congestion control. Users can select any of the following values from the drop-down list: DIAMETER_UNABLE_TO_COMPLY, DIAMETER_TOO_BUSY
Discard Policy.Danger of Congested State.Discard Priority	NA	Editable	Specifies the discard priority for the load shedding rule. The discard priority value can be a number from 0 to 15. Any request message with equal or lower priority is rejected.
Discard Policy.Danger of Congested State.Return ResultCode	NA	enum	Specifies the result code, returned in the answer response, when request message is rejected as part of congestion control. Users can select any of the following values from the drop-down list: DIAMETER_UNABLE_TO_COMPLY, DIAMETER_TOO_BUSY

Editing Diameter Gateway Congestion Configurations

To update or modify any configuration information related to diameter gateway congestion:

1. Click **Edit**. The **Edit Diameter Gateway Congestion Configurations** page appears:
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully." appears.
3. If you do not want to modify any configuration, click **Cancel**. The Diameter Gateway Congestion Configurations page appears.
4. Click **Refresh** to reload the Diameter Gateway Congestion Configurations page.

3.13.5 Provisioning Data Repository Service

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Provisioning Data Repository Service**. The **Provisioning Data Repository Service Configurations** page appears:

Figure 3-33 Provisioning Data Repository Service Configurations

Field Name	Default Value	Attribute	Description
Hikari Pool Size:	25	Read-only	The Hikari pool connection size to be created at start up.
Tracing Enabled:	false	Read-only	By default, tracing is not enabled.
On Demand Migration Enabled:	false	Editable	By default, on demand migration is not enabled.
Prov Logs Enabled:	false	Editable	Flag to enable or disable the provisioning logs.
Prov Logs API Names Supported:	nudr-dr-prov	Editable	Supported API for provisioning logs. The supported provisioning APIs are as follows: - nudr-dr-prov - nudr-group-id-map-prov - slf-group-prov - n5g-eir-prov

The following table explains the **Provisioning Data Repository Service Configurations** page in detail:

Table 3-43 Provisioning Data Repository Service Configurations Parameters

Field Name	Default Value	Attribute	Description
Hikari Pool Size	25	Read-only	The Hikari pool connection size to be created at start up.
Tracing Enabled	False	Read-only	By default, tracing is not enabled.
On Demand Migration Enabled	False	Editable	By default, on demand migration is not enabled.
Prov Logs Enabled	False	Editable	Flag to enable or disable the provisioning logs.
Prov Logs API Names Supported	nudr-dr-prov	Editable	Supported API for provisioning logs. The supported provisioning APIs are as follows: - nudr-dr-prov - nudr-group-id-map-prov - slf-group-prov - n5g-eir-prov

Editing Provisioning Data Repository Service Configurations

To update or modify any configuration information related to Provisioning Data Repository Service:

1. Click **Edit**. The **Edit Provisioning Data Repository Service Configurations** page appears.
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully." appears.
3. If you do not want to modify any configuration, click **Cancel**. The Provisioning Data Repository Service Configurations page appears.
4. Click **Refresh** to reload the Provisioning Data Repository Service Configurations page.

3.13.6 Notify Service

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Notify Service**. The **Notify Service Configurations** page appears:

Figure 3-34 Notify Service Configurations

Notify Service Configurations
Edit
Refresh

Notify Retry Count:	Notify Retry Count 0
Notify Retry Interval:	Notify Retry Interval 5
Retry Error Codes:	Retry Error Codes 400,429,500,503
Hikari Pool Size:	Hikari Pool Size 10
Tracing Enabled:	Tracing Enabled false
Resource Removal Notification Enabled:	Resource Removal Notification Enabled true

The following table explains the **Notify Service Configurations** page in detail:

Table 3-44 Notify Service Configurations Parameters

Field Name	Default Value	Attribute	Description
Notify Retry Count	3	Editable	Number of notification attempts to be done in case of notification failures. Retries are based on notification.retryerrorcodes configuration.
Notif Retry Interval	5	Editable	The retry interval for notifications in case of failure. It is measured in seconds. Retries are based on notification.retryerrorcodes configuration.
Retry Error Codes	400,429,500,503	Editable	This field displays comma separated error codes. These error codes are eligible for retry notifications in case of failures.
Hikari Pool Size	25	Read-only	The Hikari pool connection size to be created at start up.
Tracing Enabled	false	Read-only	This enables tracing on jaeger. By default, tracing is not enabled.
Resource Removal Notification Enabled	true	Editable	It is the PCF notifications for delete subscriber and deletion of policy data. When enabled, the "delResources" attribute/key is included in the notification payload.

Editing Notify Service Configurations

To update or modify any configuration information related to notify service:

1. Click **Edit**. The **Edit Notify Service Configurations** page appears.
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully," appears.
3. If you do not want to modify any configuration, click **Cancel**. The Notify Service Configurations page appears.
4. Click **Refresh** to reload the Notify Service Configurations page.

3.13.7 On Demand Migration

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **On Demand Migration**. The **On Demand Migration** page appears:

Figure 3-35 On Demand Migration Configurations

On Demand Migration Configurations

Edit Refresh Help

On Demand Enabled: On Demand Enabled
false

Delete Subscriber on Source UDR : Delete Subscriber on Source UDR
false

Realm: Realm
oracle.com

Identity: Identity
ondemand-migration.nudr.oracle.com

Source UDR Configuration

REST Endpoint: REST Endpoint
http://source-udr-rest-endpoint

Realm: Realm
tekelec1.com

Identity: Identity
local1.tekelec1.com

Host Port: Host Port
3868

Host IP: Host IP

The following table explains the **On Demand Migration Configurations** page in detail:

Table 3-45 On Demand Migration Configuration Parameters

Field Name	Default Value	Attribute	Description
On Demand Enabled	false	Read-only	Flag to enable on-demand migration for migrating subscriber from 4G UDR

Table 3-45 (Cont.) On Demand Migration Configuration Parameters

Field Name	Default Value	Attribute	Description
Delete Subscriber on Source UDR	False	Editable	Flag to delete subscriber data when it is successfully migrated from target UDR to 5G UDR
Realm	NA	Editable	Diameter realm of the target UDR
Identity	ondemand-migration.udr.oracle.com	Read-only	Diameter client identity in ondemand-migration tool
Source UDR Configuration: REST Endpoint	NA	Editable	Target UDR rest endpoint for deleting subscriber when the deleteSubscriber flag is enabled
Realm	udr.oracle.com	Read-only	Diameter client realm in ondemand-migration tool
Identity	ondemand-migration.udr.oracle.com	Read-only	Diameter client identity in ondemand-migration tool
Host Port	NA	Editable	Port number of target UDR where diameter service is running
Host IP	NA	Editable	IP address of target UDR

Editing On Demand Migration Configurations

To update or modify any configuration information related to on demand migration:

1. Click **Edit**. The **On Demand Migration Configurations** page appears.
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully," appears.
3. If you do not want to modify any configuration, click **Cancel**. You navigate back to the On Demand Migration Configurations page.
4. Click **Refresh** to reload the On Demand Migration Configurations page.

3.13.8 Subscriber Export Tool Configurations

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Subscriber Export Tool Configurations**. The **Subscriber Export Tool Configurations** page appears:

Figure 3-36 Subscriber Export Tool Configurations

Subscriber Export Tool Configurations Edit Refresh

Frequency Of Periodic Export:	Frequency Of Periodic Export ONE TIME
Export Start Time:	Export Start Time
Maximum Number of Export Dumps:	Maximum Number of Export Dumps 5
Export File Name Suffix:	Export File Name Suffix SLFTestSegment
Multi Valued Field Delimiter:	Multi Valued Field Delimiter
Field Delimiter:	Field Delimiter ,
File Extension:	File Extension .txt
IMSI Count:	IMSI Count 5
MSISDN Count:	MSISDN Count 5
Remote Path:	Remote Path /home/cloud-user/ocudr/
Delete Local Copy:	Delete Local Copy true
Subscriber Range Enabled:	Subscriber Range Enabled false
New File Per Subscriber Range Enabled:	New File Per Subscriber Range Enabled false
Exported Files Expiry Time:	Exported Files Expiry Time 7d

▼ Subscriber Range Support for IMSI MSISDN

Key Type:	Key Type MSISDN
Subscriber Ranges:	Subscriber Ranges 1111111130-1111111140,1111111150-1111111160
Policy Data Resources for CSV Export:	Policy Data Resources for CSV Export sm-data,am-data,ue-policy-set

The following table explains the parameters in the **Subscriber Export Tool Configurations** page in detail:

Table 3-46 Subscriber Export Tool Configurations Parameters

Field Name	Default Value	Attribute	Description
Frequency Of Periodic Export	DAILY	Editable	It is the frequency of periodic export of subscriber data. This is configurable as ONE_TIME/DAILY/WEEKLY/MONTHLY
Export Start Time	14/09/2022 17:32:10	Editable	Time at which periodic export starts. 24 hours format is used.
Maximum Number of Export Dumps	5	Editable	Maximum number of export
Export File Name Suffix	SLFTestSegment	Editable	The prefix name of the exported file
Multi Valued Field Delimiter		Editable	This separator is used to differentiate multiple values for a single key type.
Field Delimiter	,	Editable	Delimiter for fields
File Extension	.csv	Editable	Extension for file. This can be .csv/.txt

Table 3-46 (Cont.) Subscriber Export Tool Configurations Parameters

Field Name	Default Value	Attribute	Description
IMSI Count	5	Editable	Number of IMSI for EIR export. If 5 IMSIs are not provisioned for a given IMEI, then 0 value is supplied for remaining IMSI .
MSISDN Count	5	Editable	Number of MSISDN for EIR export. If 5 MSISDNs are not provisioned for a given IMEI, then 0 value is supplied for remaining MSISDN.
Remote Path	/home/cloud-user/	Editable	Path in remote server where the exported files is stored.
Delete Local Copy	true	Editable	Flag to enable or disable deletion of files from SFTP.
Subscriber Range Enabled	false	Editable	Flag to enable or disable the range based subscriber export. Default is set to false for full range of 4G policy subscriber data export.
New File Per Subscriber Range Enabled	False	Editable	When enabled creates a separate EXML file for each range. Default is set to false.
Exported Files Expiry Time	7d	Editable	The exported file expiry time, after which the file is deleted.
Key Type	IMSI	Editable	To export the range based subscriber data with keys. The two key type are IMSI and MSISDN.
Subscriber Ranges	NA	Editable	Configures the start and end ranges that is given to export the data. This works only if subscriberRangeEnabled parameter is enabled.
Policy Data Resource for CSV Export	It is the type of data that needs to be exported for the policy data. By default, profile data is included.	Editable	sm-data, am-data, ue-policy-set

Editing Subscriber Export Tool Configurations

To update or modify configuration information related to data repository:

1. Click **Edit**. The **Edit Subscriber Export Tool Configurations** page appears.
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully" appears.
3. If you do not want to modify any configuration, click **Cancel**. You can navigate back to the Subscriber Export Tool Configurations page.
4. Click **Refresh** to reload the Subscriber Export Tool Configurations page.

3.13.9 Migration Tool Configurations

Click **UDR** and then **Service Configurations**. From the **Service Configurations** menu, select **Migration Tool Configurations**. The **Migration Tool Configurations** page appears:

Figure 3-37 Migration Tool Configurations

The screenshot shows the 'Migration Tool Configurations' page with the following settings:

Field Name	Value
Enable Migration Pause:	Enable Migration Pause true
Key Type:	Key Type msisdn
Delete 4G subscriber:	Delete 4G subscriber false
Migration Realm:	Migration Realm oracle.com
Migration Identity:	Migration Identity migration.oracle.com
Migration Node Port:	Migration Node Port 3868
Migration Node Host:	Migration Node Host ocudr-nudr-migration-service

The following table explains the parameters in the **Migration Tool Configurations** page in detail:

Table 3-47 Migration Tool Configurations Parameters

Field Name	Description
Enable Migration Pause	If it is set to true the migration process is temporarily paused. Default Value: false Attribute: Editable
Key Type	Specifies the type of key used in the migration process. The two key types are IMSI and MSISDN. Default Value: MSISDN Attribute: Read-only
Delete 4G subscriber	Boolean field that determines whether 4G data should be deleted. Default Value: false Attribute: Read-only

Table 3-47 (Cont.) Migration Tool Configurations Parameters

Field Name	Description
Migration Realm	Represents the migration tool realm. Default Value: oracle.com Attribute: Read-only
Migration Identity	Represents the migration tool identity. Default Value: migration.oracle.com Attribute: Read-only
Migration Node Port	Represents the port number used for the migration node. Default Value: 3868 Attribute: Read-only
Migration Node Host	Specifies the host name or network address of the migration node. Default Value: ocudr-nudr-migration-service Attribute: Read-only

Editing Migration Tool Configurations

To update or modify configuration information related to data repository:

1. Click **Edit**. The **Edit Migration Tool Configurations** page appears.
2. Update the configuration details as required and click **Save**. A confirmation message, "Save successfully." appears.
3. If you do not want to modify any configuration, click **Cancel**. You can navigate back to the Migration Tool Configurations page.
4. Click **Refresh** to reload the Migration Tool Configurations page.

3.14 NF Scoring Configurations

You can configure the NF Scoring feature using the CNC Console. To navigate to NF Scoring, click **UDR** and then **NF Scoring** in the left navigation pane of the CNC Console application. Using **NF Scoring**, SLF users can configure following services:

- **Settings**
- **Calculated Score**

Settings:

From the **NF Scoring** menu, select **Settings**. The **Settings** page appears:

Settings [Edit] [Refresh]

Enable NF Scoring: Enable NF Scoring: true

TPS

- Enable: true
- Max Score: 20
- Max TPS: 20

Service Health

- Enable: true
- Max Score: 30

Signalling Connections

- Enable: true
- Max Score: 20
- Max Connections: 20

Replication Health

- Enable: true
- Max Score: 30

Locality/ Site Preference

- Enable: true
- Score: 5

Active Alert

- Enable: true
- Critical Alert Weightage: 2
- Major Alert Weightage: 1
- Minor Alert Weightage: 0

Custom Criteria:

Name	Enable	Metric Name	Scoring Type	Maximum Score	Maximum Value	Weight Age	Add
No data to display.							

The following table explains the parameters in the **Settings** page in detail:

Table 3-48 NF Scoring Settings for SLF

Field Name	Description
Enable NF Scoring	Specifies whether to enable or disable the NF Scoring. This is a mandatory parameter. Data Type: Boolean Default Value: false Range: true or false
TPS	Specifies the Transaction Per Second.
Enable	Enables the TPS. Data Type: Boolean Default Value: false Range: true or false

Table 3-48 (Cont.) NF Scoring Settings for SLF

Field Name	Description
Max Score	Specifies the maximum score of the TPS. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 20 Range: 1 to 100.
Max TPS	Specifies the maximum TPS. If enable is set to true then this is a mandatory parameter. Data Type: Integer Range: 1 to 1000000
Service Health	Specifies the Service Health of a site.
Enable	Enables the Service Health. This is a mandatory parameter. Data Type: Boolean Default Value: false Range: true or false
Max Score	Specifies the maximum score of the Service Health. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 30 Range: 1 to 100
Signaling Connections	Specifies the Signaling Connections of a site.
Enable	Enables the Signaling Connections. This is a mandatory parameter. Data Type: Boolean Default Value: false Range: true or false
Max Score	Specifies the maximum score of the Signaling Connections. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 20 Range: 1 to 100
Max Connections	Specifies the maximum connections. If enable is set to true then this is a mandatory parameter. Data Type: Integer Range: 1 to 10000
Replication Health	Specifies the Replication Health of a site.
Enable	Enables the Replication Health. Data Type: Boolean Default Value: false Range: true or false

Table 3-48 (Cont.) NF Scoring Settings for SLF

Field Name	Description
Max Score	Specifies the maximum score of the Replication Health. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 30 Range: 1 to 100
Locality or Site preference	Specifies the Locality or Site Preference.
Enable	Enables the Locality or Site Preference. Data Type: Boolean Default Value: false Range: true or false
Score	Specifies the score of the Locality or Site Preference. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 5 Range: 1 to 100
Active Alert	Specifies the Active Alerts of a site.
Enable	Enables the Active Alert. Data Type: Boolean Default Value: false Range: true or false
Critical Alert Weightage	The site with more critical alerts is unhealthy. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 2 Range: 0 to 100
Major Alert Weightage	The site with more major alerts is unhealthy. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 1 Range: 0 to 100
Minor Alert Weightage	The site with more minor alerts is unhealthy. If enable is set to true then this is a mandatory parameter. Data Type: Integer Default Value: 0 Range: 0 to 100
Custom Criteria	Specifies the Custom Criteria of a site.
Enable	Specifies whether to enable or disable the Custom Criteria. Data Type: Boolean Default Value: false Range: true or false

Table 3-48 (Cont.) NF Scoring Settings for SLF

Field Name	Description
Name	Name of the Custom Criteria. The two applicable criteria are Subscriber Bulk Import tool and Subscriber Export tool. Data Type: String
Metric Name	Specifies the name of the Metric. Data Type: String
Scoring Type	Specifies the scoring type. The two scoring types are ratio and weightage. Data Type: String
Max Score	Specifies the maximum score. Data Type: Integer Default Value: 30 Range: 1 to 100
Max Value	Specifies the maximum value. Data Type: Integer Range: 0 to 10000
Weightage	Specifies the weightage of the alert. Data Type: Integer Default Value: 1 Range: 1 to 100

To update or modify the configuration information related to **Settings**:

1. Click **Edit**. The Edit **Settings** page appears.
2. To add the **Custom Criteria**, click **Add**.
 - a. The **Add Custom Criteria** dialog box appears.

 **Add Custom Criteria** 

Name:

Enable:

☐

Metric Name:

Scoring Type:

Scoring Type

Ratio



Maximum Score:

Required

Maximum Value:

Required

Save

Cancel

✕

Add Custom Criteria

Name:

Name

Enable:

☐

Metric Name:

Metric Name

Scoring Type:

Scoring Type
Weightage

Weightage:

Weightage

Required

Save

Cancel

b.

Note

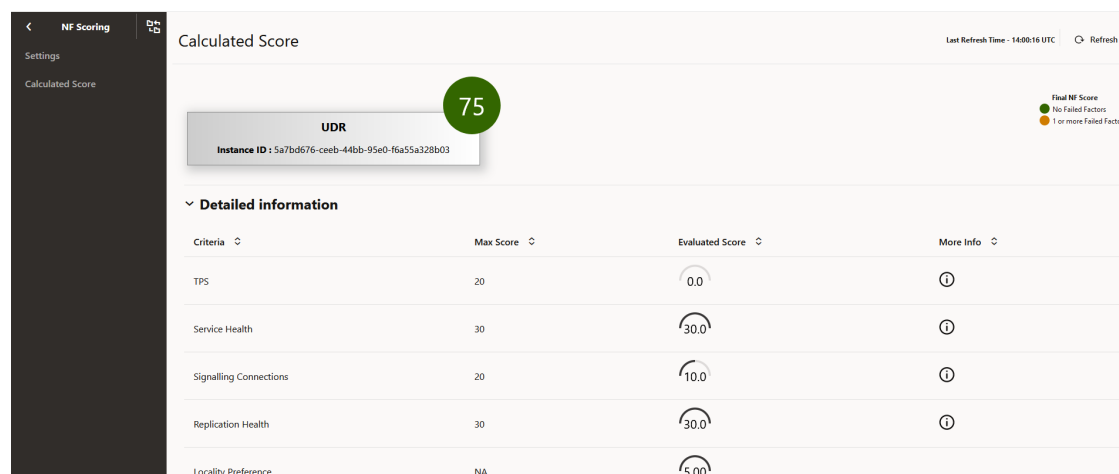
If the scoring type is selected as weightage then maximum score and minimum score fields are disabled.

Fill in the required details and click **Save** to add the custom criteria.

- c. Click **Cancel** to exit the **Add Custom Criteria** dialog box.
3. Update the configuration details as required in the settings page and click **Save**. A confirmation message, "Save successfully." appears.
4. If you do not want to modify any configuration, click **Cancel**.
5. Click **Refresh** to reload the **Settings** page.

Calculated Score:

From the **NF Scoring** menu, select **Calculated Score**. If the NF Scoring feature is enabled. The **Calculated Score** page appears as below:

Figure 3-38 Calculated Score**Note**

If app-info pod is down, you will not be getting the NF Score. You will get an error message that "Data can't be fetched due to internal server error".

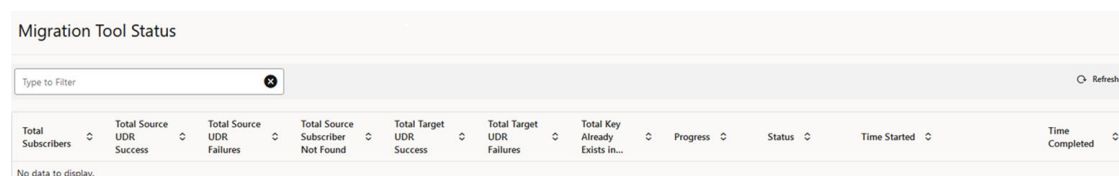
Calculated Score shows the total score along with the Instance ID. The total score is shown in either Green or Orange color. If the NF Score is shown in green color then there are no failed factors. And, if the NF Score is shown in Orange color there are one or more failed factors.

You can click on **Detailed information** to view different criteria and their Max Score, Evaluated Score, and More Info. The criteria in the detailed information tab show the evaluated score. The failed factors are shown with a warning symbol under the evaluated score. On the top-right of the screen, the Last Refresh Time information is available and a **Refresh** button is given to refresh the NF Score of a site.

3.15 Migration Tool Status

The Migration Tool Status page is a monitoring page, which is read-only. This page allows the operator to monitor the Migration Tool Status, which initiates on running the Migration Tool. For more information, see [Migration Tool](#).

To access the Migration Tool Status page, click **UDR** and then, **Migration Tool Status** in the left navigation pane of the CNC Console application. The following page appears:

Figure 3-39 Migration Tool Status Page

The Migration Tool Status page shows following transaction details:

- **Total Subscribers:** The total number of subscribers included in the migration process.
- **Total Source UDR Success:** The total number of records successfully migrated from 4G UDR.
- **Total Source UDR Failure:** The total number of records failed during the migration from 4G UDR.
- **Total Source Subscriber Not Found:** The total number of subscribers not found for migration from 4G UDR.
- **Total Target UDR Success:** The total number of records successfully migrated to 5G UDR.
- **Total Target UDR Failure:** The total number of record failed during the migration in 5G UDR.
- **Total Key Already Exists in the Target UDR:** The total number of keys that already exist in 5G UDR.
- **Progress:** Shows migration progress completion in percentage.
- **Status:** The overall status of the migration process. For example, **STARTED**, **IN PROGRESS**, **PAUSED**, or **COMPLETED**.
- **Time Started:** Shows the migration process start time.
- **Time Completed:** Shows the migration process completion time.

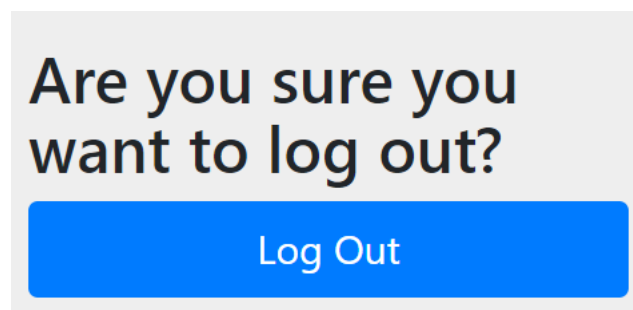
Note

Using **Filter** option, you can filter the records you are interested in and using **Refresh** option, you can refresh the Subscriber Bulk Import Status page.

3.16 Logout of CNC Console

To logout of the CNC Console application, click the **Sign Out** link available at the top right corner. The following confirmation message appears:

Figure 3-40 CNC Console Logout



Click **Log Out**. The user exits the application and the CNC Console login page appears.

4

Alerts

Oracle Communications Cloud Native Core, Unified Data Repository (UDR) uses alerts to track following scenarios:

- Pod not running or down
- Pod restarts
- Transaction reaches maximum threshold traffic
- Subscriber not found
- XFCC validation failure rate
- Invalid user agent

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.

If any of the above scenarios occur, an alert triggers in Prometheus. Alerts help to handle a scenario before its failure.

Configuring the Alerts Manually

Use the following files to configure the alerts manually. The files are shared as part of custom templates of UDR:

- **ocudr_alerts_haprom.yaml** to configure alerts in UDR when using CNE 25.1.2xx version.
- **ocudr_alerts_non_haprom.yaml** to configure alerts in UDR when using environment other than CNE.
- **ocslf_alerts_non_haprom.yaml** to configure alerts in SLF when using OSO 25.1.2xx versions.
- **ocslf_alerts_haprom.yaml** to configure alerts in SLF when using CNE 25.1.2xx version.
- **oceir_alerts_haprom.yaml** to configure alerts in EIR when using CNE 25.1.2xx version.
- **oceir_alerts_non_haprom.yaml** to configure alerts in EIR when using environment other than CNE.

Manually Configuring the Alerts in SLF when using OSO 25.1.2xx versions:

In the SLF_Alerts.yaml file update the namespace and then perform the following steps to configure alerts manually:

1. Run the following command to take backup of current configuration map of Prometheus.

```
kubectl get configmaps occne-prometheus-server -o yaml -n occne-infra
> /tmp/tempConfig.yaml
```

2. Run the following commands to add UDR alerts file to the Prometheus configmap yaml file.

```
sed -i '/etc\config\alertsudr\d' /tmp/tempConfig.yaml
```

```
sed -i '/rule_files:/a\ \ - /etc/config/alertsudr' /tmp/tempConfig.yaml
```

3. Run the following command to update the configuration map with the updated file name of SLF alert file.

```
kubectl replace configmap occne-prometheus-server -f /tmp/tempConfig.yaml
```

4. Run the following command to add UDR_Alertrules in the configuration map under the SLF alert file name.

```
kubectl patch configmap occne-prometheus-server -n occne-infra --type
merge --patch "$(cat ~/ocslf_alerts_non_haprom_25.1.200.yaml)"
```

Note

Prometheus server takes updated configmap automatically after sometime (approximately 20 sec).

Manually Configuring the Alerts in UDR, SLF, and EIR when using CNE 25.1.2xx version:

In the UDR, SLF, and EIR Alerts.yaml, update the namespace and then perform the following steps to configure the alerts manually:

Run the following command for:

UDR:

```
kubectl create -f ocudr_alerts_haprom_25.1.200.yaml -n <namespace>
```

SLF:

```
kubectl create -f ocslf_alerts_haprom_25.1.200.yaml -n <namespace>
```

EIR:

```
kubectl create -f oceir_alerts_haprom_25.1.200.yaml -n <namespace>
```

Manually Disabling the Alerts

Perform the following steps to disable the alerts manually in the Prometheus:

1. Edit the `ocslf_alerts_non_haprom.yaml` file to remove specific alert. For example, to disable the `OcudrTrafficRateAboveMinorThreshold` alert:

```
## ALERT SAMPLE START##

- alert: OcudrTrafficRateAboveMinorThreshold
  annotations:
    description: 'Ingress traffic Rate is above minor threshold i.e.
800 mps (current value is: {{ $value }})'
    summary: 'Traffic Rate is above 80 Percent of Max requests per
second(1000)'
    expr:
sum(rate(oc_ingressgateway_http_requests_total{app_kubernetes_io_name="ingr
essgateway",kubernetes_namespace="ocudr"}[2m])) >= 800 < 900
    labels:
      severity: Minor

## ALERT SAMPLE END##
```

2. Remove the specific content from alert that needs to be disabled.
3. Configure alert again. For more information, see the [To Configure](#) section.

OSO Alerts Automation

Alerts are automated by using Helm upgrade command with the Helm chart provided as part of OSO software package. The alert automation process is as follows:

1. A new `oso-alr-config` Helm chart is provided as part of OSO software package from 25.1.200 release onwards. For information to download OSO software package, see *Oracle Communications, Cloud Native Core, Operations Services Overlay Installation and Upgrade Guide*.
2. The `oso-alr-config` Helm chart must be deployed once OSO is installed.
3. This separate Helm chart allows the Helm install command to run without an input alert file.

```
helm install oso-alr-config ocoso_csar_abc_25_1_200_0_0-
rc.1_alert_config_charts.tgz -f ocoso_csar_abc_25_1_200_0_0-
rc.1_alert_config_custom_values.yaml -n <oso deployed namespace>
```

4. When the `oso-alr-config` Helm chart installation is completed then the `oso-alr-config` is ready to use.
5. Run Helm upgrade, if you are enabling this feature after UDR deployment. Run the following Helm upgrade command in `oso-alr-config` file to apply UDR alert file.

```
helm upgrade oso-alr-config oso-alr-config/ -f ocoso_csar_abc_25_1_200_0_0-
rc.1_alert_config_custom_values.yaml -f
ocslf_alerts_non_haprom_25.1.200.yaml -n <oso deployed namespace>
```

6. Once the Helm upgrade is completed, you can view the alerts file that is applied to OSO **Prometheus ConfigMap**. This can be viewed in the Prometheus Graphical User Interface (GUI).
7. You can also update the changes in the same alert file and perform a Helm upgrade. The alert file will be updated with the latest changes.

Perform the following steps to clear the alerts:

1. An empty `ocsf_alertrules_empty_<version>.yaml` file is delivered as part of OSO software package. For information to download OSO software package, see *Oracle Communications, Cloud Native Core, Operations Services Overlay Installation and Upgrade Guide*. You must provide this `ocsf_alertrules_empty_<version>.yaml` file during the Helm upgrade.
2. This `ocsf_alertrules_empty_<version>.yaml` file is used to remove all the alerts using the Helm upgrade command by providing `ocsf_alertrules_empty_<version>.yaml` file as an input file. This removes the alerts from the OSO **Prometheus ConfigMap** and **Prometheus GUI** and keeps the references under `rule_files "/etc/config/alertsudir"` and the alert rules will be empty `"alertsudir: { }"`.
3. For example, a sample Helm upgrade command to clean up alert rules is as follows:

```
helm upgrade oso-alr-config oso-alr-config/ -f ocoso_csar_abc_25_1_200_0_0-rc.1_alert_config_custom_values.yaml -f
ocsf_alertsrules_empty_25.1.200.yaml -n <oso deployed namespace>
```

Sample empty alert file is as follows:

```
apiVersion: v1
data:
  alertsudir: |
    {}
```

Observe

For more information on metrics and KPIs, see [UDR Metrics](#) and [UDR KPIs](#) sections respectively.

4.1 Alert Details

This section describes alerts in detail.

① Note

Max Ingress requests/sec in consideration is 1000/second.

Table 4-1 Alerts Levels or Severity Types

Alerts Levels / Severity Types	Definition
Critical	Indicates a severe issue that poses a significant risk to safety, security, or operational integrity. It requires immediate response to address the situation and prevent serious consequences. Raised for conditions may affect the service of UDR.
Major	Indicates a more significant issue that has an impact on operations or poses a moderate risk. It requires prompt attention and action to mitigate potential escalation. Raised for conditions may affect the service of UDR.

Table 4-1 (Cont.) Alerts Levels or Severity Types

Alerts Levels / Severity Types	Definition
Minor	Indicates a situation that is low in severity and does not pose an immediate risk to safety, security, or operations. It requires attention but does not demand urgent action. Raised for conditions may affect the service of UDR.
Info or Warn (Informational)	Provides general information or updates that are not related to immediate risks or actions. These alerts are for awareness and do not typically require any specific response. WARN and INFO alerts may not impact the service of UDR.

The below table provides alert names for UDR and EIR.

Table 4-2 Alert names for UDR/SLF and EIR

UDR/SLF	EIR
OcudrTrafficRateAboveMajorThreshold	OceirTrafficRateAboveMajorThreshold
OcudrTrafficRateAboveMinorThreshold	OceirTrafficRateAboveMinorThreshold
OcudrTrafficRateAboveCriticalThreshold	OceirTrafficRateAboveCriticalThreshold
OcudrTransactionErrorRateAbove0.1Percent	OceirTransactionErrorRateAbove0.1Percent
OcudrTransactionErrorRateAbove1Percent	OceirTransactionErrorRateAbove1Percent
OcudrTransactionErrorRateAbove10Percent	OceirTransactionErrorRateAbove10Percent
OcudrTrafficRateAboveCriticalThreshold	OceirTrafficRateAboveCriticalThreshold
OcudrTrafficRateAboveMajorThreshold	OceirTrafficRateAboveMajorThreshold
OcudrTrafficRateAboveMinorThreshold	OceirTrafficRateAboveMinorThreshold
OcudrTransactionErrorRateAbove0.1Percent	OceirTransactionErrorRateAbove0.1Percent
OcudrTransactionErrorRateAbove1Percent	OceirTransactionErrorRateAbove1Percent
OcudrTransactionErrorRateAbove10Percent	OceirTransactionErrorRateAbove10Percent
OcudrTransactionErrorRateAbove25Percent	OceirTransactionErrorRateAbove25Percent
OcudrTransactionErrorRateAbove50Percent	OceirTransactionErrorRateAbove50Percent
OcudrSubscriberNotFoundAbove1Percent	OceirSubscriberNotFoundAbove1Percent
OcudrSubscriberNotFoundAbove10Percent	OceirSubscriberNotFoundAbove10Percent
OcudrSubscriberNotFoundAbove25Percent	OceirSubscriberNotFoundAbove25Percent
OcudrSubscriberNotFoundAbove50Percent	OceirSubscriberNotFoundAbove50Percent
OcudrPodsRestart	OceirPodsRestart
NudrServiceDown	NudrServiceDown
NudrProvServiceDown	NudrProvServiceDown
NudrNotifyServiceServiceDown	NA
NudrNRFCClientServiceDown	NudrNRFCClientServiceDown
NudrConfigServiceDown	NudrConfigServiceDown
NudrDiameterProxyServiceDown	NudrDiameterProxyServiceDown

Table 4-2 (Cont.) Alert names for UDR/SLF and EIR

UDR/SLF	EIR
NudrOnDemandMigrationServiceDown	NA
OcudrIngressGatewayServiceDown	OceirIngressGatewayServiceDown
OcudrEgressGatewayServiceDown	OceirEgressGatewayServiceDown
OcudrDbServiceDown	OceirDbServiceDown
OcudrXFCCValidationFailureAbove10Percent	OceirXFCCValidationFailureAbove10Percent
OcudrXFCCValidationFailureAbove20Percent	OceirXFCCValidationFailureAbove20Percent
OcudrXFCCValidationFailureAbove50Percent	OceirXFCCValidationFailureAbove50Percent
DRServiceOverload60Percent	DRServiceOverload60Percent
DRServiceOverload75Percent	DRServiceOverload75Percent
DRServiceOverload80Percent	DRServiceOverload80Percent
DRServiceOverload90Percent	DRServiceOverload90Percent
SLFSucessTxnDefaultGroupIdRateAbove1Percent	NA
SLFSucessTxnDefaultGroupIdRateAbove10Percent	NA
SLFSucessTxnDefaultGroupIdRateAbove25Percent	NA
SLFSucessTxnDefaultGroupIdRateAbove50Percent	NA
OcudrDiameterCongestionCongestedState	OceirDiameterCongestionCongestedState
OcudrDiameterCongestionDocState	OceirDiameterCongestionDocState
DRProvServiceOverload60Percent	DRProvServiceOverload60Percent
DRProvServiceOverload75Percent	DRProvServiceOverload75Percent
DRProvServiceOverload80Percent	DRProvServiceOverload80Percent
DRProvServiceOverload90Percent	DRProvServiceOverload90Percent
OcudrIngressGatewayProvServiceDown	OceirIngressGatewayProvServiceDown
OcudrProvisioningTrafficRateAboveMajorThreshold	OceirProvisioningTrafficRateAboveMajorThreshold
OcudrProvisioningTrafficRateAboveCriticalThreshold	OceirProvisioningTrafficRateAboveCriticalThreshold
OcudrProvisioningTransactionErrorRateAbove25Percent	OceirProvisioningTransactionErrorRateAbove25Percent
OcudrProvisioningTransactionErrorRateAbove50Percent	OceirProvisioningTransactionErrorRateAbove50Percent
PVCFullForSLFExport	NA
FailedExtractForSLFExport	NA
BulkImportTransferInFailed	BulkImportTransferInFailed
BulkImportTransferOutFailed	BulkImportTransferOutFailed
ExportToolTransferOutFailed	ExportToolTransferOutFailed
PVCFullForXMLBulkImport	PVCFullForXMLBulkImport
PVCFullForBulkImport	PVCFullForBulkImport
OperationalStatusCompleteShutdown	OperationalStatusCompleteShutdown
NFScoreCalculationFailed	NFScoreCalculationFailed

Table 4-2 (Cont.) Alert names for UDR/SLF and EIR

UDR/SLF	EIR
PVCFullForUDRExport	NA
UDRExportFailed	NA
IngressgatewayPodProtectionDocState	IngressgatewayPodProtectionDocState
IngressgatewayPodProtectionCongestedState	IngressgatewayPodProtectionCongestedState
RetryNotificationRecordsMaxLimitExceeded	RetryNotificationRecordsMaxLimitExceeded
UserAgentHeaderNotFoundMorethan10PercentRequest	NA
EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold	EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold
EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold	EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold
EgressGatewayJVMBufferMemoryUsedAboveCriticalThreshold	EgressGatewayJVMBufferMemoryUsedAboveCriticalThreshold
NudrDiameterGatewayDown	NudrDiameterGatewayDown
DiameterPeerConnectionsDropped	DiameterPeerConnectionsDropped
IGWSignallingPodProtectionDOCState	NA
IGWSignallingPodProtectionCongestedState	NA
IGWSignallingPodProtectionByRateLimitRejectedRequest	NA

Note

For the following alert details, only UDR alerts names are provided. The corresponding EIR alert names can be found in [Table 4-2](#).

4.1.1 System Level Alerts

This section lists the system level alerts.

4.1.1.1 OcudrSubscriberNotFoundAbove1Percent

Table 4-3 OcudrSubscriberNotFoundAbove1Percent

Field	Details
Description	Total number of response if subscriber not found is about 1% of ingress traffic
Summary	Total number of response if subscriber not found is about 1% of ingress traffic
Severity	Warning
Condition	Alert if number of subscribers not found is 1% of all ingress traffic
OID	1.3.6.1.4.1.323.5.3.43.1.2.7009
Metric Used	udr_subscriber_not_found_total

Table 4-3 (Cont.) OcudrSubscriberNotFoundAbove1Percent

Field	Details
Recommended Actions	The alert is cleared when the number of failure of Subscriber Not Found are below 1% of the total. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.1.2 OcudrSubscriberNotFoundAbove10Percent

Table 4-4 OcudrSubscriberNotFoundAbove10Percent

Field	Details
Description	Total number of response if subscriber not found is about 10% of ingress traffic
Summary	Total number of response if subscriber not found is about 10% of ingress traffic
Severity	Minor
Condition	Alert if number of subscribers not found is 10% of all ingress traffic
OID	1.3.6.1.4.1.323.5.3.43.1.2.7010
Metric Used	udr_subscriber_not_found_total
Recommended Actions	The alert is cleared when the number of failure of Subscriber Not Found are below 10% of the total. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.1.3 OcudrSubscriberNotFoundAbove25Percent

Table 4-5 OcudrSubscriberNotFoundAbove25Percent

Field	Details
Description	Total number of response if subscriber not found is about 25% of ingress traffic
Summary	Total number of response if subscriber not found is about 25% of ingress traffic
Severity	Major
Condition	Alert if number of subscribers not found is 25% of all ingress traffic
OID	1.3.6.1.4.1.323.5.3.43.1.2.7011
Metric Used	udr_subscriber_not_found_total

Table 4-5 (Cont.) OcudrSubscriberNotFoundAbove25Percent

Field	Details
Recommended Actions	The alert is cleared when the number of failure of Subscriber Not Found are below 25% of the total. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: <code>udr_rest_failure_response_total</code> 2. If guidance required, Contact My Oracle Support.

4.1.1.4 OcudrSubscriberNotFoundAbove50Percent

Table 4-6 OcudrSubscriberNotFoundAbove50Percent

Field	Details
Description	Total number of response if subscriber not found is about 50% of ingress traffic
Summary	Total number of response if subscriber not found is about 50% of ingress traffic
Severity	Critical
Condition	Alert if number of subscribers not found is 50% of all ingress traffic
OID	1.3.6.1.4.1.323.5.3.43.1.2.7012
Metric Used	<code>udr_subscriber_not_found_total</code>
Recommended Actions	The alert is cleared when the number of failure of Subscriber Not Found are below 50% of the total. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: <code>udr_rest_failure_response_total</code> 2. If guidance required, Contact My Oracle Support.

4.1.1.5 OcudrPodsRestart

Table 4-7 OcudrPodsRestart

Field	Details
Description	Pod <code>{{ \$labels.pod }}</code> has restarted.
Summary	namespace: <code>{{ \$labels.namespace }}</code> , podname: <code>{{ \$labels.kubernetes_pod_name }}</code> , timestamp: <code>{{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}</code> : A Pod has restarted
Severity	Major
Condition	Alert if any of the pod got restarted
OID	1.3.6.1.4.1.323.5.3.43.1.2.7014
Metric Used	<code>kube_pod_container_status_restarts_total</code>

Table 4-7 (Cont.) OcudrPodsRestart

Field	Details
Recommended Actions	The alert is cleared automatically if the specific pod is up. Steps: 1. Refer to the application logs on Kibana and filter based on pod name, check for database related failures such as connectivity, kubernetes secrets and so on. 2. Check orchestration logs for liveness or readiness probe failures using the following commands. <code>kubectrl get po -n <namespace></code> Note the full name of the pod that is not running and use it in the following command. <code>kubectrl describe pod <desired full pod name> -n <namespace></code> 3. Check the DB status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. For more information, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

4.1.1.6 NudrServiceDown

Table 4-8 NudrServiceDown

Field	Details
Description	OCUDR Nudr_DRService {{{labels.app_kubernetes_io_name}}} is down
Summary	namespace: {{{labels.kubernetes_namespace}}}, podname: {{{labels.kubernetes_pod_name}}}, timestamp: {{{ with query "time()" }}} . first value humanizeTimestamp }}} end }} : DR Service is down
Severity	Critical
Condition	Alert if Nudr-dr service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7015
Metric Used	app_kubernetes_io_name="nudr-dr-service

Table 4-8 (Cont.) NudrServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NudrService service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.7 NudrProvServiceDown

Table 4-9 NudrProvServiceDown

Field	Details
Description	OCUDR Nudr_DR_PROVService {{\$labels.app_kubernetes_io_name}} is down
Summary	'namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : DR Prov Service is down'
Severity	Critical
Condition	Alert if Nudr-dr service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7015
Metric Used	app_kubernetes_io_name="nudr-dr-provservice

Table 4-9 (Cont.) NudrProvServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NudrProvService service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.8 NudrNotifyServiceServiceDown

Table 4-10 NudrNotifyServiceServiceDown

Field	Details
Description	OCUDR NudrNotifyServiceService {{\$labels.app_kubernetes_io_name}} is down
Summary	namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : Nudr Notify Service down.
Severity	Critical
Condition	Alert if Nudr Notify service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7016
Metric Used	app_kubernetes_io_name="nudr-notify-service"

Table 4-10 (Cont.) NudrNotifyServiceServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NotifyService service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.9 NudrNRFClientServiceDown

Table 4-11 NudrNRFClientServiceDown

Field	Details
Description	OCUDR NRFClient service {{\$labels.app_kubernetes_io_name}} is down
Summary	namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : NRF Client service down
Severity	Critical
Condition	Alert if Nudr Nrf Client service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7017
Metric Used	app_kubernetes_io_name="nrf-client-nfmanagement

Table 4-11 (Cont.) NudrNRFCClientServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NRFCClientService service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.10 NudrConfigServiceDown

Table 4-12 NudrConfigServiceDown

Field	Details
Description	OCUDR config service {{labels.app_kubernetes_io_name}} is down
Summary	namespace: {{labels.kubernetes_namespace}}, podname: {{labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : nudr-config service down
Severity	Critical
Condition	Alert if Nudr Config service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7020
Metric Used	app_kubernetes_io_name="nudr-config"

Table 4-12 (Cont.) NudrConfigServiceDown

Field	Details
Recommended Actions	The alert is cleared when the ConfigService service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.11 NudrDiameterProxyServiceDown

Table 4-13 NudrDiameterProxyServiceDown

Field	Details
Description	OCUDR diameterproxy service {{\$labels.app_kubernetes_io_name}} is down
Summary	namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : nudr-diameterproxy service is down
Severity	Critical
Condition	Alert if Nudr Diameter Proxy is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7018
Metric Used	app_kubernetes_io_name="nudr-diameterproxy"

Table 4-13 (Cont.) NudrDiameterProxyServiceDown

Field	Details
Recommended Actions	The alert is cleared when the DiameterProxyService service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.12 NudrOnDemandMigrationServiceDown

Table 4-14 NudrOnDemandMigrationServiceDown

Field	Details
Description	OCUDR ondemand-migration service {{labels.app_kubernetes_io_name}} is down
Summary	namespace: {{labels.kubernetes_namespace}}, podname: {{labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : NFSubscription service is down
Severity	Critical
Condition	Alert if Nudr On Demand Migration is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7019
Metric Used	app_kubernetes_io_name="nudr-ondemand-migration"

Table 4-14 (Cont.) NudrOnDemandMigrationServiceDown

Field	Details
Recommended Actions	The alert is cleared when the OnDemandMigrationService service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. <code>kubectrl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectrl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.13 OcudrIngressGatewayServiceDown

Table 4-15 OcudrIngressGatewayServiceDown

Field	Details
Description	OCUDR Ingress-Gateway service {{\$labels.app_kubernetes_io_name}} is down
Summary	namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : Ingress-gateway service down
Severity	Critical
Condition	Alert if Ingress Service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7021
Metric Used	app_kubernetes_io_name="ingressgateway"

Table 4-15 (Cont.) OcudrIngressGatewayServiceDown

Field	Details
Recommended Actions	The alert is cleared when the ingressgateway service is available. Steps: 1. Check the orchestration logs of ingress-gateway service and check for liveness or readiness probe failures using the following commands. <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on ingress-gateway service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.14 OcudrEgressGatewayServiceDown

Table 4-16 OcudrEgressGatewayServiceDown

Field	Details
Description	OCUDR Egress-Gateway service {{\$labels.app_kubernetes_io_name}} is down
Summary	namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : Egress-Gateway service down
Severity	Critical
Condition	Alert if Egress Service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7022
Metric Used	app_kubernetes_io_name="egressgateway"

Table 4-16 (Cont.) OcudrEgressGatewayServiceDown

Field	Details
Recommended Actions	The alert is cleared when the egressgateway service is available. Note: The threshold is configurable in the UDR_Alertrules.yaml Steps: 1. Check the orchestration logs of egress-gateway service and check for liveness or readiness probe failures using the following commands. <code>kubect! get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command <code>kubect! describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on egress-gateway service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and Contact My Oracle Support. Note: Use CNC NF Data Collector tool for capturing logs. Refer "NF Data Collector tool user guide" for more details.

4.1.1.15 OcudrDbServiceDown

Table 4-17 OcudrDbServiceDown

Field	Details
Description	Mysql connectivity service is down
Summary	namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : MySQL connectivity service down
Severity	Critical
Condition	Alert if Mysql connectivity is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7023
Metric Used	appinfo_service_running
Recommended Actions	This alert clears when the microservice nudr-drservice is up and running.

4.1.1.16 OcudrIngressGatewayProvServiceDown

Table 4-18 OcudrIngressGatewayProvServiceDown

Field	Details
Description	OCUDR Ingress-Gateway service {{\$labels.app_kubernetes_io_name}} is down

Table 4-18 (Cont.) OcudrIngressGatewayProvServiceDown

Field	Details
Summary	namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : Ingress-gateway service down
Severity	Critical
Condition	Alert if Ingressgateway-prov service is down
OID	1.3.6.1.4.1.323.5.3.43.1.2.7043
Metric Used	app_kubernetes_io_name="ingressgateway-prov"
Recommended Actions	The alert is cleared when the ingress-gateway service is available. Steps: 1. Check the orchestration logs of the ingress-gateway service and check for liveness or readiness probe failures using the following commands: <code>kubectl get po -n <namespace></code> Note the full name of the pod that is not running. It must be used in the following command: <code>kubectl describe pod <specific desired full pod name> -n <namespace></code> 2. Refer the application logs on Kibana and filter based on the ingress-gateway service names. Check for the ERROR WARNING logs related to the thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support. Note Use the CNC NF Data Collector tool for capturing logs. Refer to NF Data Collector tool user guide for more details.

4.1.2 Application Level Alerts

This section lists the application level alerts.

4.1.2.1 OcudrTrafficRateAboveMajorThreshold

Table 4-19 OcudrTrafficRateAboveMajorThreshold

Field	Details
Description	'Ingress traffic Rate is above major threshold i.e. 900 requests per second
Summary	'Traffic Rate is above 90 Percent of Max requests per second(1000)'
Severity	Major
Condition	Alert if Ingress traffic reaches 90% of max TPS

Table 4-19 (Cont.) OcudrTrafficRateAboveMajorThreshold

Field	Details
OID	1.3.6.1.4.1.323.5.3.43.1.2.7002
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. Note: The threshold is configurable in the UDR_Alertrules.yaml Steps: Reassess why the OCUDR is receiving additional traffic (eg : Mated site OCUDR is unavailable in georedundancy scenario). If this is unexpected, contact My Oracle Support and: 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress Gateway section in Grafana to determine an increase in 4xx and 5xx error codes. 3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.

4.1.2.2 OcudrTrafficRateAboveMinorThreshold

Table 4-20 OcudrTrafficRateAboveMinorThreshold

Field	Details
Description	Ingress traffic rate is above minor threshold i.e. 800 requests per second
Summary	Traffic rate is above 80 Percent of Max requests per second(1000)
Severity	Minor
Condition	Alert if Ingress traffic reaches 80% of max TPS
OID	1.3.6.1.4.1.323.5.3.43.1.2.7001
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	The alert is cleared either when the total Ingress Traffic rate falls below the Minor threshold or when the total traffic rate cross the Major threshold, in which case the OcudrTrafficRateAboveMinorThreshold alert shall be raised. Note: The threshold is configurable in the UDR_Alertrules.yaml Steps: Reassess why the OCUDR is receiving additional traffic(eg : Mated site OCUDR is unavailable in geo redundancy scenario). If this is unexpected, contact My Oracle Support and: 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress Gateway section in Grafana to determine increase in 4xx and 5xx Error codes. 3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.

4.1.2.3 OcudrTrafficRateAboveCriticalThreshold

Table 4-21 OcudrTrafficRateAboveCriticalThreshold

Field	Details
Description	'Ingress traffic Rate is above critical threshold i.e. 950 requests per second
Summary	'Traffic Rate is above 95 Percent of Max requests per second(1000)'
Severity	Critical
Condition	Alert if Ingress traffic reaches 95% of max TPS
OID	1.3.6.1.4.1.323.5.3.43.1.2.7003
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. Note: The threshold is configurable in the UDR_Alertrules.yaml Steps: Reassess why the OCUDR is receiving additional traffic (Example: Mated site OCUDR is unavailable in geo redundancy scenario). If this is unexpected, contact My Oracle Support and: 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress Gateway section in Grafana to determine increase in 4xx and 5xx Error codes. 3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.

4.1.2.4 OcudrTransactionErrorRateAbove0.1Percent

Table 4-22 OcudrTransactionErrorRateAbove0.1Percent

Field	Details
Description	Transaction error rate is above 0.1 Percent of Total Transactions
Summary	Transaction Error Rate detected above 0.1 Percent of Total Transactions
Severity	Warning
Condition	Alert if all error rate exceeds 0.1% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7004
Metric Used	oc_ingressgateway_http_responses_total

Table 4-22 (Cont.) OcudrTransactionErrorRateAbove0.1Percent

Field	Details
Recommended Actions	The alert is cleared when the number of failed transactions is below 0.1 percent of the total transactions or when the number of failed transactions crosses the 1% threshold in which case the OcudrTransactionErrorRateAbove0.1Percent is raised. Steps: 1. Check metrics per service, per method For example, discovery requests can be deduced from these metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance is required, Contact My Oracle Support.

4.1.2.5 OcudrTransactionErrorRateAbove1Percent

Table 4-23 OcudrTransactionErrorRateAbove1Percent

Field	Details
Description	'Transaction Error rate is above 1 Percent of Total Transactions'
Summary	'Transaction Error Rate detected above 1 Percent of Total Transactions'
Severity	Warning
Condition	Alert if all error rate exceeds 1% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7005
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions are below 1% of the total transactions or when the number of failure transactions cross the 10% threshold in which case the OcnrfTransactionErrorRateAbove10Percent shall be raised. Steps: 1. Check metrics per service, per method For example discovery requests can be deduced from this metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, Contact My Oracle Support.

4.1.2.6 OcudrTransactionErrorRateAbove10Percent

Table 4-24 OcudrTransactionErrorRateAbove10Percent

Field	Details
Description	Transaction error rate is above 10 Percent of Total Transactions
Summary	Transaction Error Rate detected above 10 Percent of Total Transactions

Table 4-24 (Cont.) OcudrTransactionErrorRateAbove10Percent

Field	Details
Severity	Minor
Condition	Alert if all error rate exceeds 10% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7006
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions are below 10% of the total transactions or when the number of failure transactions cross the 25% threshold in which case the OcnrfTransactionErrorRateAbove25Percent shall be raised. Steps: 1. Check metrics per service, per method For example discovery requests can be deduced from this metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, Contact My Oracle Support.

4.1.2.7 OcudrTrafficRateAboveCriticalThreshold

Table 4-25 OcudrTrafficRateAboveCriticalThreshold

Field	Details
Description	Ingress traffic rate is above critical threshold i.e. 950 requests per second
Summary	Traffic rate is above 95 Percent of Max requests per second(1000)
Severity	Critical
Condition	Alert if Ingress traffic reaches 95% of max TPS
OID	1.3.6.1.4.1.323.5.3.43.1.2.7003
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. Note: The threshold is configurable in the UDR_Alertrules.yaml Steps: Reassess why the OCUDR is receiving additional traffic (Example: Mated site OCUDR is unavailable in geo redundancy scenario). If this is unexpected, contact My Oracle Support and: 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress Gateway section in Grafana to determine increase in 4xx and 5xx Error codes. 3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.

4.1.2.8 OcudrTrafficRateAboveMajorThreshold

Table 4-26 OcudrTrafficRateAboveMajorThreshold

Field	Details
Description	Ingress traffic rate is above major threshold i.e. 900 requests per second
Summary	Traffic rate is above 90 Percent of Max requests per second(1000)
Severity	Major
Condition	Alert if Ingress traffic reaches 90% of max TPS
OID	1.3.6.1.4.1.323.5.3.43.1.2.7002
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. Note: The threshold is configurable in the UDR_Alertrules.yaml Steps: Reassess why the OCUDR is receiving additional traffic (eg: Mated site OCUDR is unavailable in geo redundancy scenario). If this is unexpected, contact My Oracle Support and: 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine increase in 4xx and 5xx Error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

4.1.2.9 OcudrTrafficRateAboveMinorThreshold

Table 4-27 OcudrTrafficRateAboveMinorThreshold

Field	Details
Description	Ingress traffic Rate is above minor threshold i.e. 800 requests per second
Summary	Traffic Rate is above 80 Percent of Max requests per second (1000)
Severity	Minor
Condition	Alert if Ingress traffic reaches 80% of max TPS
OID	1.3.6.1.4.1.323.5.3.43.1.2.7001
Metric Used	oc_ingressgateway_http_requests_total

Table 4-27 (Cont.) OcudrTrafficRateAboveMinorThreshold

Field	Details
Recommended Actions	The alert is cleared either when the total Ingress Traffic rate falls below the Minor threshold or when the total traffic rate cross the Major threshold, in which case the OcudrTrafficRateAboveMinorThreshold alert shall be raised. Note: The threshold is configurable in the UDR_Alertrules.yaml Steps: Reassess why the OCUDR is receiving additional traffic (eg : Mated site OCUDR is unavailable in geo redundancy scenario). If this is unexpected, contact My Oracle Support and: 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress Gateway section in Grafana to determine increase in 4xx and 5xx Error codes. 3. Check Ingress Gateway logs on Kibana to determine the reason for the errors.

4.1.2.10 OcudrTransactionErrorRateAbove0.1Percent

Table 4-28 OcudrTransactionErrorRateAbove0.1Percent

Field	Details
Description	Transaction Error rate is above 0.1 Percent of Total Transactions
Summary	Transaction Error Rate detected above 0.1 Percent of Total Transactions
Severity	Warning
Condition	Alert if all error rate exceeds 0.1% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7004
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions are below 0.1 percent of the total transactions or when the number of failure transactions cross the 1% threshold in which case the OcudrTransactionErrorRateAbove0.1Percent shall be raised. Steps: 1. Check metrics per service, per method For example discovery requests can be deduced from this metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, Contact My Oracle Support.

4.1.2.11 OcudrTransactionErrorRateAbove1Percent

Table 4-29 OcudrTransactionErrorRateAbove1Percent

Field	Details
Description	Transaction Error rate is above 1 Percent of Total Transactions
Summary	Transaction Error Rate detected above 1 Percent of Total Transactions
Severity	Warning
Condition	Alert if all error rate exceeds 1% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7005
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions are below 1% of the total transactions or when the number of failure transactions cross the 10% threshold in which case the OcnrfTransactionErrorRateAbove10Percent shall be raised. Steps: 1. Check metrics per service, per method For example discovery requests can be deduced from this metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, contact My Oracle Support.

4.1.2.12 OcudrTransactionErrorRateAbove10Percent

Table 4-30 OcudrTransactionErrorRateAbove10Percent

Field	Details
Description	Transaction Error rate is above 10 Percent of Total Transactions
Summary	Transaction Error Rate detected above 10 Percent of Total Transactions
Severity	Minor
Condition	Alert if all error rate exceeds 10% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7006
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions are below 10% of the total transactions or when the number of failure transactions cross the 25% threshold in which case the OcnrfTransactionErrorRateAbove25Percent shall be raised. Steps: 1. Check metrics per service, per method For example discovery requests can be deduced from this metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, Contact My Oracle Support.

4.1.2.13 OcudrTransactionErrorRateAbove25Percent

Table 4-31 OcudrTransactionErrorRateAbove25Percent

Field	Details
Description	Transaction Error Rate detected above 25 Percent of Total Transactions
Summary	Transaction Error Rate detected above 25 Percent of Total Transactions
Severity	Major
Condition	Alert if all error rate exceeds 25% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7007
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions are below 25% of the total transactions or when the number of failure transactions cross the 50% threshold in which case the OcudrTransactionErrorRateAbove50Percent shall be raised. Steps: 1. Check metrics per service, per method For example discovery requests can be deduced from this metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, Contact My Oracle Support.

4.1.2.14 OcudrTransactionErrorRateAbove50Percent

Table 4-32 OcudrTransactionErrorRateAbove50Percent

Field	Details
Description	Transaction Error Rate detected above 50 Percent of Total Transactions
Summary	Transaction Error Rate detected above 50 Percent of Total Transactions
Severity	Critical
Condition	Alert if all error rate exceeds 50% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7008
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions are below 50 percent of the total transactions. Steps: 1. Check metrics per service, per method For example, discovery requests can be deduced from this metrics Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, Contact My Oracle Support.

4.1.2.15 OcudrXFCCValidationFailureAbove10Percent

Table 4-33 OcudrXFCCValidationFailureAbove10Percent

Field	Details
Description	Total number of response with xfcc validation failure is about 10% of ingress traffic
Summary	Total number of response with xfcc validation failure is about 10% of ingress traffic
Severity	Minor
Condition	Alert if XFCC validation failure is 10% of the total XFCC validations
OID	1.3.6.1.4.1.323.5.3.43.1.2.7024
Metric Used	oc_ingressgateway_xfcc_header_validate_total
Recommended Actions	The alert is cleared when the number of failure of XFCCValidationFailure are below 10% of the total. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.2.16 OcudrXFCCValidationFailureAbove20Percent

Table 4-34 OcudrXFCCValidationFailureAbove20Percent

Field	Details
Description	Total number of response with xfcc validation failure is about 20% of ingress traffic
Summary	Total number of response with xfcc validation failure is about 20% of ingress traffic
Severity	Major
Condition	Alert if XFCC validation failure is 20% of the total XFCC validations
OID	1.3.6.1.4.1.323.5.3.43.1.2.7025
Metric Used	oc_ingressgateway_xfcc_header_validate_total
Recommended Actions	The alert is cleared when the number of failure of XFCCValidationFailure are below 20% of the total. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.2.17 OcudrXFCCValidationFailureAbove50Percent

Table 4-35 OcudrXFCCValidationFailureAbove50Percent

Field	Details
Description	Total number of response with XFCC validation failure is about 50% of ingress traffic
Summary	Total number of response with XFCC validation failure is about 50% of ingress traffic.
Severity	Critical
Condition	Alert if XFCC validation failure is 50% of the total XFCC validations
OID	1.3.6.1.4.1.323.5.3.43.1.2.7026
Metric Used	oc_ingressgateway_xfcc_header_validate_total
Recommended Actions	The alert is cleared when the number of failure of XFCCValidationFailure are below 50% of the total. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.2.18 DRServiceOverload60Percent

Table 4-36 DRServiceOverload60Percent

Field	Details
Description	This alert is fired when the application go to the overload level of Warn level
Summary	This alert is fired when the application go to the overload level of Warn level
Severity	Warning
Condition	Alert If the application overloads at 60%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7027
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below Warn level. Steps: 1. Check the service specific metrics to understand the specific service request errors. for eg: udr_rest_failure_response_total 2. If guidance required, contact My Oracle Support.

4.1.2.19 DRServiceOverload75Percent

Table 4-37 DRServiceOverload75Percent

Field	Details
Description	This alert is fired when the application go to the overload level of Minor level
Summary	This alert is fired when the application go to the overload level of Minor level.
Severity	Minor
Condition	Alert If the application overloads at 75%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7028
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below Minor level. Steps: 1. Check the service specific metrics to understand the specific service request errors. for eg: udr_rest_failure_response_total 2. If guidance required, contact My Oracle Support.

4.1.2.20 DRServiceOverload80Percent

Table 4-38 DRServiceOverload80Percent

Field	Details
Description	This alert is fired when the application go to the overload level of Minor level
Summary	This alert is fired when the application go to the overload level of Minor level
Severity	Major
Condition	Alert If the application overloads at 80%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7029
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below Major level. Steps: 1. Check the service specific metrics to understand the specific service request errors. for eg: udr_rest_failure_response_total 2. If guidance required, contact My Oracle Support.

4.1.2.21 DRServiceOverload90Percent

Table 4-39 DRServiceOverload90Percent

Field	Details
Description	This alert is fired when the application go to the overload level of Minor level
Summary	This alert is fired when the application go to the overload level of Minor level
Severity	Critical
Condition	Alert if the application overloads at 90%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7030
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below Critical level. Steps: 1. Check the service specific metrics to understand the specific service request errors. for eg: udr_rest_failure_response_total 2. If guidance required, contact My Oracle Support.

4.1.2.22 SLFSucessTxnDefaultGroupldRateAbove1Percent

Table 4-40 SLFSucessTxnDefaultGroupldRateAbove1Percent

Field	Details
Description	Transaction Error Rate detected above 1 Percent of Total Transactions
Summary	Transaction Error rate is above 1 Percent of Total Transactions
Severity	Warning
Condition	Alert if number of SLF Lookup requests responded with default Group ID exceeds 1% of the total responses.
OID	1.3.6.1.4.1.323.5.3.43.1.2.7031
Metric Used	slf_sucess_txn_default_grp_id_total
Recommended Actions	This alert is cleared when SLF Lookup request coming for subscribers not provisioned reduces. Steps: Check the subscriber range received for Lookup and make sure to avoid if there is any unexpected out of range of subscribers.

4.1.2.23 SLFSucessTxnDefaultGroupldRateAbove10Percent

Table 4-41 SLFSucessTxnDefaultGroupldRateAbove10Percent

Field	Details
Description	Transaction Error Rate detected above 10 Percent of Total Transactions
Summary	Transaction Error rate is above 10 Percent of Total Transactions
Severity	Minor

Table 4-41 (Cont.) SLFSucessTxnDefaultGroupldRateAbove10Percent

Field	Details
Condition	Alert if number of SLF Lookup requests responded with default Group ID exceeds 10% of the total responses.
OID	1.3.6.1.4.1.323.5.3.43.1.2.7032
Metric Used	slf_sucess_txn_default_grp_id_total
Recommended Actions	This alert is cleared when SLF Lookup request coming for subscribers not provisioned reduces. Steps: Check the subscriber range received for Lookup and make sure to avoid if there is any unexpected out of range of subscribers.

4.1.2.24 SLFSucessTxnDefaultGroupldRateAbove25Percent

Table 4-42 SLFSucessTxnDefaultGroupldRateAbove25Percent

Field	Details
Description	Transaction Error Rate detected above 25 Percent of Total Transactions
Summary	Transaction Error rate is above 25 Percent of Total Transactions
Severity	Major
Condition	Alert if number of SLF Lookup requests responded with default Group ID exceeds 25% of the total responses.
OID	1.3.6.1.4.1.323.5.3.43.1.2.7033
Metric Used	slf_sucess_txn_default_grp_id_total
Recommended Actions	This alert is cleared when SLF Lookup request coming for subscribers not provisioned reduces. Steps: Check the subscriber range received for Lookup and make sure to avoid if there is any unexpected out of range of subscribers.

4.1.2.25 SLFSucessTxnDefaultGroupldRateAbove50Percent

Table 4-43 SLFSucessTxnDefaultGroupldRateAbove50Percent

Field	Details
Description	Transaction Error Rate detected above 50 Percent of Total Transactions
Summary	Transaction Error rate is above 50 Percent of Total Transactions
Severity	Critical
Condition	Alert if number of SLF Lookup requests responded with default Group ID exceeds 50% of the total responses.
OID	1.3.6.1.4.1.323.5.3.43.1.2.7034
Metric Used	slf_sucess_txn_default_grp_id_total

Table 4-43 (Cont.) SLFSucessTxnDefaultGroupIdRateAbove50Percent

Field	Details
Recommended Actions	This alert is cleared when SLF Lookup request coming for subscribers not provisioned reduces. Steps: Check the subscriber range received for Lookup and make sure to avoid if there is any unexpected out of range of subscribers.

4.1.2.26 OcudrDiameterCongestionCongestedState

Table 4-44 OcudrDiameterCongestionCongestedState

Field	Details
Description	Alert will be raised if the diameter gateway pod is in CONGESTED state.
Summary	Alert will be raised if the diameter gateway pod is in CONGESTED state.
Severity	Critical
Condition	Alert will be raised if the diameter gateway pod is in CONGESTED state.
Metric Used	ocudr_pod_congestion_state = = 2
Recommended Actions	This alert is raised when the Diameter Gateway pod congestion level is set to the CONGESTED state. Steps: 1. Decrease the traffic run or use proper perf resource. 2. Check the pod congestion configurations and resource limit in CNC Console.

4.1.2.27 OcudrDiameterCongestionDocState

Table 4-45 OcudrDiameterCongestionDocState

Field	Details
Description	Alert will be raised if the diameter gateway pod is in is in Danger of Congestion (DOC) state.
Summary	Alert will be raised if the diameter gateway pod is in is in Danger of Congestion (DOC) state.
Severity	Major
Condition	Alert will be raised if the diameter gateway pod is in is in Danger of Congestion (DOC) state.
Metric Used	ocudr_pod_congestion_state = = 1
Recommended Actions	This alert is raised when the Diameter Gateway pod congestion level is set to the Danger of Congestion (DOC) state. Steps: 1. Decrease the traffic run or use proper perf resource. 2. Check the pod congestion configurations and resource limit in CNC Console.

4.1.2.28 DRProvServiceOverload60Percent

Table 4-46 DRProvServiceOverload60Percent

Field	Details
Description	This alert is fired when the application go to the overload level of Warn level
Summary	This alert is fired when the application go to the overload level of Warn level
Severity	Warning
Condition	Alert If the application overloads at 60%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7036
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below Warn level. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.2.29 DRProvServiceOverload75Percent

Table 4-47 DRProvServiceOverload75Percent

Field	Details
Description	This alert is fired when the application go to the overload level of Minor level
Summary	This alert is fired when the application go to the overload level of Minor level
Severity	Minor
Condition	Alert If the application overloads at 75%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7037
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below Minor level. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.2.30 DRProvServiceOverload80Percent

Table 4-48 DRProvServiceOverload80Percent

Field	Details
Description	This alert is fired when the application go to the overload level of Major level
Summary	This alert is fired when the application go to the overload level of Major level
Severity	Major
Condition	Alert If the application overloads at 80%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7038
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below Major level. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.2.31 DRProvServiceOverload90Percent

Table 4-49 DRProvServiceOverload90Percent

Field	Details
Description	This alert is fired when the application go to the overload level of critical level
Summary	This alert is fired when the application go to the overload level of critical level
Severity	Critical
Condition	Alert If the application overloads at 90%
OID	1.3.6.1.4.1.323.5.3.43.1.2.7039
Metric Used	load_level
Recommended Actions	This alert is cleared when the incoming traffic is reduced to below critical level. Steps: 1. Check the Service specific metrics to understand the specific service request errors. Example: udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

4.1.2.32 Diameter-Gateway pod congestion Danger of congestion state

Table 4-50 Diameter-Gateway pod congestion Danger of congestion state

Field	Details
Description	DiameterGateway pod at Danger of Congestion state
Summary	DiameterGateway pod at Danger of Congestion state
Severity	Major
Condition	Alert if the diameter gateway pod is in Danger of Congestion (DOC) state
OID	1.3.6.1.4.1.323.5.3.43.1.2.7041
Metric Used	occpn_pod_congestion_state==1
Recommended Actions	This alert is raised when the diameter gateway pod congestion level is set to the danger of congestion(DOC) Steps: 1. Decrease the traffic run or use proper perf resource. 2. Make sure the pod congestion configurations and resource limit in CNE GUI.

4.1.2.33 Diameter-Gateway pod CONGESTED state

Table 4-51 Diameter-Gateway pod CONGESTED state

Field	Details
Description	DiameterGateway pod at Congested state
Summary	DiameterGateway pod at Congested state
Severity	Critical
Condition	Alert if the diameter gateway pod is in CONGESTED state
OID	1.3.6.1.4.1.323.5.3.43.1.2.7042
Metric Used	occpn_pod_congestion_state==2
Recommended Actions	This alert is raised when the diameter gateway pod congestion level is set to the CONGESTED state Steps: 1. Decrease the traffic run or use proper perf resource. 2. Make sure the pod congestion configurations and resource limit in CNE GUI

4.1.2.34 OcudrProvisioningTrafficRateAboveMajorThreshold

Table 4-52 OcudrProvisioningTrafficRateAboveMajorThreshold

Field	Details
Description	Ingress traffic Rate is above critical threshold, that is, 950 requests per second
Summary	Traffic Rate is above 95 Percent of Max requests per second (1000)

Table 4-52 (Cont.) OcudrProvisioningTrafficRateAboveMajorThreshold

Field	Details
Severity	Critical
Condition	Alert if Ingress traffic reaches 95% of max TPS
OID	1.3.6.1.4.1.323.5.3.43.1.2.7044
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. Note The threshold is configurable in <code>UDR_Alertrules.yaml</code>. Steps: Reassess why OCUDR is receiving an additional traffic (for example, Mated site OCUDR is unavailable in geo redundancy scenario). If this is unexpected, contact My Oracle Support. 1. Refer Grafana to determine the service that is receiving high traffic. 2. Refer to the Ingress gateway section in Grafana to determine an increase in 4xx and 5xx Error codes. 3. Check the Ingress gateway logs on Kibana to determine the reason for the errors.

4.1.2.35 OcudrProvisioningTrafficRateAboveCriticalThreshold

Table 4-53 OcudrProvisioningTrafficRateAboveCriticalThreshold

Field	Details
Description	Ingress traffic Rate is above major threshold, that is, 900 requests per second
Summary	Traffic Rate is above 90 Percent of Max requests per second (1000)
Severity	Major
Condition	Alert if Ingress traffic reaches 90% of max TPS
OID	1.3.6.1.4.1.323.5.3.43.1.2.7045
Metric Used	oc_ingressgateway_http_requests_total

Table 4-53 (Cont.) OcudrProvisioningTrafficRateAboveCriticalThreshold

Field	Details
Recommended Actions	The alert is cleared when the total Ingress Traffic rate falls below the Major threshold or when the total traffic rate exceeds the Critical threshold in which the OcudrTrafficRateAboveMajorThreshold alert is raised. Note The threshold is configurable in <code>UDR_Alertrules.yaml</code>. Steps: Reassess why OCUDR is receiving an additional traffic (for example, Mated site OCUDR is unavailable in geo redundancy scenario). If this is unexpected, contact My Oracle Support. 1. Refer Grafana to determine the service that is receiving high traffic. 2. Refer to the Ingress gateway section in Grafana to determine an increase in 4xx and 5xx Error codes. 3. Check the Ingress gateway logs on Kibana to determine the reason for the errors.

4.1.2.36 OcudrProvisioningTransactionErrorRateAbove25Percent

Table 4-54 OcudrProvisioningTransactionErrorRateAbove25Percent

Field	Details
Description	Transaction Error Rate detected above 25 Percent of Total Transactions
Summary	Transaction Error Rate detected above 25 Percent of Total Transactions
Severity	Major
Condition	Alert if all error rate exceeds 25% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7046
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions is below 25% of the total transactions or when the number of failure transactions exceeds the 50% threshold in which the OcudrTransactionErrorRateAbove50Percent is raised. Steps: 1. Check the metrics per service per method, for example, discovery requests can be deduced from these metrics. Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, contact My Oracle Support.

4.1.2.37 OcudrProvisioningTransactionErrorRateAbove50Percent

Table 4-55 OcudrProvisioningTransactionErrorRateAbove50Percent

Field	Details
Description	Transaction Error Rate detected above 50 Percent of Total Transactions
Summary	Transaction Error Rate detected above 50 Percent of Total Transactions
Severity	Critical
Condition	Alert if all error rate exceeds 50% of the total transactions
OID	1.3.6.1.4.1.323.5.3.43.1.2.7047
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	The alert is cleared when the number of failure transactions is below 50 percent of the total transactions. Steps: 1. Check the metrics per service per method, for example, discovery requests can be deduced from these metrics. Metrics="oc_ingressgateway_http_responses_total" Method="GET" Status="503 SERVICE_UNAVAILABLE" 2. If guidance required, contact My Oracle Support.

4.1.2.38 PVCFullForSLFExport

Table 4-56 PVCFullForSLFExport

Field	Details
Description	Storage for Export tool is full
Summary	Storage for Export tool is full
Severity	Critical
Condition	Alert if PVC allocated for export tool dump path is full
Metric Used	export_tool_full_usage
Recommended Actions	Alert will be cleared when the PVC usage is optimized. Configure maxDumps to lower value to clear old dumps. Remove old dumps, if any from the export tool container.

4.1.2.39 FailedExtractForSLFExport

Table 4-57 FailedExtractForSLFExport

Field	Details
Description	Export tool job is failed
Summary	Export tool job is failed
Severity	Critical
Condition	Alert of the export operation fails
Metric Used	export_failure

Table 4-57 (Cont.) FailedExtractForSLFExport

Field	Details
Recommended Actions	Check logs for failure. The alert will be cleared when the export job succeeds next time.

4.1.2.40 BulkImportTransferInFailed

Table 4-58 BulkImportTransferInFailed

Field	Details
Description	Transfer-in failed for bulk import
Summary	Transfer-in failed for bulk import
Severity	Major
Condition	Alert will be raised, if Transfer-In failed from Remote to PVC
Metric Used	bulkimport_transfer_in_status
Recommended Actions	This alert is cleared when the transfer-in is success from bulk import. Steps 1. Check the service specific metrics to understand the specific service request errors. For example, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

4.1.2.41 ExportToolTransferOutFailed

Table 4-59 ExportToolTransferOutFailed

Field	Details
Description	Transfer-out failed for export-tool
Summary	Transfer-out failed for export-tool"
Severity	Major
Condition	Alert will be raised if Transfer-Out failed from PVC to Remote
Metric Used	sftp_transfer_status
Recommended Actions	This alert is cleared when the transfer-out is success from export tool. Steps 1. Check the service specific metrics to understand the specific service request errors.. For example, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

4.1.2.42 BulkImportTransferOutFailed

Table 4-60 BulkImportTransferOutFailed

Field	Details
Description	Transfer-out failed for bulk import
Summary	Transfer-out failed for bulk import
Severity	Major
Condition	Alert will be raised if Transfer-Out failed from PVC to Remote
Metric Used	bulkimport_transfer_out_status
Recommended Actions	This alert is cleared when the transfer-out is success from bulk import. Steps 1. Check the service specific metrics to understand the specific service request errors. For example, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

4.1.2.43 PVCFullForXMLBulkImport

Table 4-61 PVCFullForXMLBulkImport

Field	Details
Description	Storage for XML Bulk Import tool is full
Summary	Storage for XML Bulk Import tool is full
Severity	Critical
Condition	Alert will be raised if the PVC is full for xml-csv container
Metric Used	nudr_bulk_import_tool_pvc_full_usage{app_kubernetes_io_name="nudr-xmltocsv",kubernetes_namespace="ocudr"}==1
Recommended Actions	This alert will be cleared when the PVC is back to normal. Steps: 1. Check the service specific metrics to understand the specific service request errors. For example, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

4.1.2.44 PVCFullForBulkImport

Table 4-62 PVCFullForBulkImport

Field	Details
Description	Storage for Bulk Import tool is full
Summary	Storage for Bulk Import tool is full
Severity	Critical
Condition	Alert will be raised if the PVC is full for bulk import container
Metric Used	nudr_bulk_import_tool_pvc_full_usage{app_kubernetes_io_name="nudr-bulk-import",kubernetes_namespace="ocudr"}==1

Table 4-62 (Cont.) PVCFullForBulkImport

Field	Details
Recommended Actions	This alert will be cleared when the PVC is back to normal. Steps: 1. Check the service specific metrics to understand the specific service request errors. For example, <code>udr_rest_failure_response_total</code>. 2. Contact My Oracle Support, if guidance is required.

4.1.2.45 OperationalStatusCompleteShutdown

Table 4-63 OperationalStatusCompleteShutdown

Field	Details
Description	Operational state is control shutdown
Summary	Operational state is control shutdown
Severity	Critical
Condition	Alert will be raised if the operational state of the UDR, SLF, or EIR is COMPLETE_SHUTDOWN
Metric Used	<code>nudr_config_operational_status{kubernetes_namespace="ocudr"}==1</code>
Recommended Actions	This alert will be cleared when the operational status is back to normal. Steps: 1. Check the service specific metrics to understand the specific service request errors. For example, <code>udr_rest_failure_response_total</code>. 2. Contact My Oracle Support, if guidance is required.

4.1.2.46 NFScoreCalculationFailed

Table 4-64 NFScoreCalculationFailed

Field	Details
Description	NFScoreCalculationFailed
Summary	NFScoreCalculationFailed
Severity	Major
Condition	Alert is raised if the NF Score calculation are failed for any of the scoring factors
Metric Used	<code>nfscore{kubernetes_namespace="ocudr",factor=~"successTPS signallingConnections serviceHealth replicationHealth localityPreference bulkImport bulkExport",calculatedStatus="failed"}</code>
Recommended Actions	This alert is cleared when the NF score calculation is successful. Steps: 1. Check the service specific metrics to understand the specific service request errors. For example, <code>udr_rest_failure_response_total</code>. 2. Contact My Oracle Support, if guidance is required.

4.1.2.47 PVCFullForUDRExport

Table 4-65 PVCFullForUDRExport

Field	Details
Description	Storage for Export tool is full
Summary	Storage for Export tool is full
Severity	Critical
Condition	Alert is raised if PVC allocated for export tool dump path is full.
Metric Used	export_tool_full_usage{namespace="ocudr"}==1
Recommended Actions	Alert is cleared when the PVC usage is optimized. You must configure maxDumps to a lower value to clear old dumps. Steps: 1. If present, remove the old dumps from the export tool container.

4.1.2.48 UDRExportFailed

Table 4-66 UDRExportFailed

Field	Details
Description	Export tool job is failed
Summary	Export tool job is failed
Severity	Critical
Condition	Alert is raised if the export operation fails for UDR Mode
Metric Used	export_failure{namespace="ocudr"}== 1
Recommended Actions	You must check the logs for failure. When the next export job is successful the alert is cleared.

4.1.2.49 IngressgatewayPodProtectionDocState

Table 4-67 IngressgatewayPodProtectionDocState

Field	Details
Description	Ingress congestion in Doc state
Summary	Ingress congestion Doc state
Severity	Critical
Condition	Alert is raised if Ingress congestion is in doc state.
Metric Used	oc_ingressgateway_pod_congestion_state{namespace="ocudr"}==1
Recommended Actions	This alert will be cleared when the ingress gateway comes to normal state. Steps: 1. Check the service specific metrics to understand the specific service request errors. For example, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

4.1.2.50 IngressgatewayPodProtectionCongestedState

Table 4-68 IngressgatewayPodProtectionCongestedState

Field	Details
Description	Ingress congestion in Congested state
Summary	Ingress congestion in Congested state
Severity	Critical
Condition	Alert is raised if ingress congestion is in congested state.
Metric Used	oc_ingressgateway_pod_congestion_state{namespace="ocudr"}==2
Recommended Actions	This alert will be cleared when the ingress gateway comes to normal state. Steps: 1. Check the service specific metrics to understand the specific service request errors. For example, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

4.1.2.51 RetryNotificationRecordsMaxLimitExceeded

Table 4-69 RetryNotificationRecordsMaxLimitExceeded

Field	Details
Description	Alert will be raised if the retry notifications stored in UDR database exceeds maximum limit.
Summary	Alert will be raised if the retry notifications stored in UDR database exceeds maximum limit.
Severity	Critical
Condition	Alert will be raised if the retry notifications stored in UDR database exceeds maximum limit.
Metric Used	nudr_notif_records_limit_exceeded{namespace="ocudr"}==1
Recommended Actions	This alert is raised when there are more notification failures and the retry notifications stored in database is more than 50k. Steps: 1. Check the notification failure rate and fix the reason for failures. This reduces the number of notifications marked for retry that is stored in UDR database. 2. Contact My Oracle Support, if guidance is required.

4.1.2.52 UserAgentHeaderNotFoundMorethan10PercentRequest

Table 4-70 UserAgentHeaderNotFoundMorethan10PercentRequest

Field	Details
Description	Alert will be raised if the total number of requests not having User-Agent header is 10% of ingress traffic when suppress notification feature is enabled.
Summary	Alert will be raised if the total number of requests not having User-Agent header is 10% of ingress traffic when suppress notification feature is enabled.
Severity	Critical
Condition	Alert will be raised if the total number of requests not having User-Agent header is 10% of ingress traffic.
Metric Used	(sum by(namespace) (rate(suppress_user_agent_not_found_total{namespace="ocudr"} [5m]))/sum by(namespace) (rate(oc_ingressgateway_http_requests_total{namespace="ocudr"} [5m])))*100 >= 10
Recommended Actions	This alert is cleared if the total number of requests not having User-Agent header is less than 10% of ingress traffic. Steps: 1. Check the service specific metrics to understand the specific service request errors. 2. Contact My Oracle Support, if guidance is required.

4.1.2.53 EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold

Table 4-71 EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold

Field	Details
Description	Alert will be raised if egress gateway JVM buffer memory is above the minor threshold limit.
Summary	Alert will be raised if egress gateway JVM buffer memory is above the minor threshold limit.
Severity	Minor
Condition	Alert will be raised if egress gateway JVM buffer memory is above the minor threshold limit.
Metric Used	sum by (id, pod) (jvm_buffer_memory_used_bytes{namespace="ocudr",pod=~".*egress.*"} [5m]) >= 1300000000
Recommended Actions	This alert is cleared if the egress gateway JVM buffer memory is below the minor threshold limit. Steps: 1. Check the reason for egress gateway JVM buffer memory is above the threshold limit. and why it is not clearing sufficient memory by itself to reach below the threshold limit. 2. Contact My Oracle Support, if guidance is required.

4.1.2.54 EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold

Table 4-72 EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold

Field	Details
Description	Alert will be raised if egress gateway JVM buffer memory is above the major threshold limit.
Summary	Alert will be raised if egress gateway JVM buffer memory is above the major threshold limit.
Severity	Major
Condition	Alert will be raised if egress gateway JVM buffer memory is above the major threshold limit.
Metric Used	sum by (id, pod) (jvm_buffer_memory_used_bytes{namespace="ocudr",pod=~".*egress.*"}) >= 1500000000
Recommended Actions	This alert is cleared if the egress gateway JVM buffer memory is below the major threshold limit. Steps: 1. Check the reason for egress gateway JVM buffer memory is above the threshold limit. and why it is not clearing sufficient memory by itself to reach below the threshold limit. 2. Contact My Oracle Support, if guidance is required.

4.1.2.55 EgressGatewayJVMBufferMemoryUsedAboveCriticalThreshold

Table 4-73 EgressGatewayJVMBufferMemoryUsedAboveCriticalThreshold

Field	Details
Description	Alert will be raised if egress gateway JVM buffer memory is above the critical threshold limit.
Summary	Alert will be raised if egress gateway JVM buffer memory is above the critical threshold limit.
Severity	Critical
Condition	Alert will be raised if egress gateway JVM buffer memory is above the critical threshold limit.
Metric Used	sum by (id, pod) (jvm_buffer_memory_used_bytes{namespace="ocudr",pod=~".*egress.*"}) >= 1800000000
Recommended Actions	This alert is cleared if the egress gateway JVM buffer memory is below the critical threshold limit. Steps: 1. Check the reason for egress gateway JVM buffer memory is above the threshold limit. and why it is not clearing sufficient memory by itself to reach below the threshold limit. 2. Contact My Oracle Support, if guidance is required.

4.1.2.56 NudrDiameterGatewayDown

Table 4-74 NudrDiameterGatewayDown

Field	Details
Description	Alert will be raised if Nudr-diam-gateway service is down.
Summary	Alert will be raised if Nudr-diam-gateway service is down.
Severity	Critical
Condition	Alert will be raised if Nudr-diam-gateway service is down.
Metric Used	<code>absent(up{container="nudr-diam-gateway",namespace="ocudr"})</code> or <code>up{container="nudr-diam-gateway",namespace="ocudr"} == 0</code>
Recommended Actions	This alert is cleared when the NudrDiamGateway service is available. Steps: - Run the following command to check the orchestration logs of appinfo service and check for liveness or readiness probe failures. <pre>kubectl get po -n <namespace></pre> - Run the following command using the full name of the pod that is not running. <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> - Refer the application logs on Kibana and filter based on the appinfo service names. Check for ERROR WARNING logs related to thread exceptions. - Perform the resolution steps depending on the reason for failure. - Contact My Oracle Support, if guidance is required. Note: Use CNC NF Data Collector tool for capturing logs. For more information, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

4.1.2.57 DiameterPeerConnectionsDropped

Table 4-75 DiameterPeerConnectionsDropped

Field	Details
Description	Alert will be raised if there are no connections between diameter peer and diameter gateway.
Summary	Alert will be raised if there are no connections between diameter peer and diameter gateway.
Severity	Major
Condition	Alert will be raised if there are no connections between diameter peer and diameter gateway.

Table 4-75 (Cont.) DiameterPeerConnectionsDropped

Field	Details
Metric Used	<code>sum(ocudr_diam_conn_network{origHost=~".*CHI.*",container="nuds-diam-gateway",namespace="ocudr"}) or vector(0) < 2 or</code> <code>sum(ocudr_diam_conn_network{origHost=~".*IND.*",container="nuds-diam-gateway",namespace="ocudr"}) or vector(0) < 2 or</code> <code>(sum(ocudr_diam_conn_network{origHost=~".*CHI.*",container="nuds-diam-gateway",kubernetes_namespace="ocudr"}) or vector(0)) +</code> <code>sum(ocudr_diam_conn_network{origHost=~".*IND.*",container="nuds-diam-gateway",namespace="ocudr"}) or vector(0) < 5</code>
Recommended Actions	This alert is cleared when the NudsDiamGateway service is available. Steps: - Run the following command to check the orchestration logs of appinfo service and check for liveness or readiness probe failures. <pre>kubectl get po -n <namespace></pre> - Run the following command using the full name of the pod that is not running. <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> - Refer the application logs on Kibana and filter based on the appinfo service names. Check for ERROR WARNING logs related to thread exceptions. - Perform the resolution steps depending on the reason for failure. - Contact My Oracle Support, if guidance is required. Note: Use CNC NF Data Collector tool for capturing logs. For more information, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

4.1.2.58 IGWSignallingPodProtectionDOCState

Table 4-76 IGWSignallingPodProtectionDOCState

Field	Details
Description	Alert will be raised when the ingress gateway signaling traffic at DOC State.
Summary	Alert will be raised when the ingress gateway signaling traffic at DOC State.
Severity	Major
Condition	Alert will be raised when the ingress gateway signaling traffic at DOC State.
Metric Used	<code>sum({namespace="ocudr",container="ingressgateway-sig"}) by (pod) == 2</code>

Table 4-76 (Cont.) IGWSignallingPodProtectionDOCState

Field	Details
Recommended Actions	This alert is cleared when the signaling traffic reaches NORMAL state. Steps: 1. Check the service specific metrics to for the specific service request errors. For example, <code>oc_ingressgateway_congestion_system_state</code>. 2. Contact My Oracle Support, if guidance is required.

4.1.2.59 IGWSignallingPodProtectionCongestedState

Table 4-77 IGWSignallingPodProtectionCongestedState

Field	Details
Description	Alert will be raised when the ingress gateway signaling traffic at Congested State.
Summary	Alert will be raised when the ingress gateway signaling traffic at Congested State.
Severity	Critical
Condition	Alert will be raised when the ingress gateway signaling traffic at Congested State.
Metric Used	<code>sum(oc_ingressgateway_congestion_system_state{namespace="ocudr",container="ingressgateway-sig"}) by (pod) == 3</code>
Recommended Actions	This alert is cleared when the signaling traffic reaches NORMAL or DOC state. Steps: 1. Check the service specific metrics to for the specific service request errors. For example, <code>oc_ingressgateway_congestion_system_state</code>. 2. Contact My Oracle Support, if guidance is required.

4.1.2.60 IGWSignallingPodProtectionByRateLimitRejectedRequest

Table 4-78 IGWSignallingPodProtectionByRateLimitRejectedRequest

Field	Details
Description	Alert will be raised when total rejections crossed more than 1% traffic of the total incoming traffic.
Summary	Alert will be raised when total rejections crossed more than 1% traffic of the total incoming traffic.
Severity	Critical
Condition	Alert will be raised when total rejections crossed more than 1% traffic of the total incoming traffic.

Table 4-78 (Cont.) IGWSignallingPodProtectionByRateLimitRejectedRequest

Field	Details
Metric Used	(sum (rate(oc_ingressgateway_http_request_ratelimit_denied_count_total{Action="REJECT",namespace="ocudr"}[2m]) or (up * 0))) / sum(rate(oc_ingressgateway_http_requests_total{container="ingressgateway-sig",namespace="ocudr"}[2m])) * 100 >= 1
Recommended Actions	This alert is cleared when the when rejection is reduced less than 1% of the total traffic. Steps: 1. Check the service specific metrics to for the specific service request errors. For example, oc_ingressgateway_congestion_system_state. 2. Contact My Oracle Support, if guidance is required.

5

KPIs

This section provides information about the KPIs for Oracle Communications Unified Data Repository (OCUDR).

Note

A sample OCUDR dashboard is provided for Grafana. To achieve KPIs, OCUDR requires metrics and functions, which are available at [Metrics](#). You can download and import the file to your repository. <http://docs.oracle.com/en/industries/communications/cloud-native-core/release21.html>

5.1 Common KPIs for UDR, SLF, and EIR

Following table provides the common KPI details UDR, SLF, and EIR.

Table 5-1 Prov Success(2xx) Rate

Field	Details
Description	Measures the success rate percentage for the provisioning traffic.
Expression	<code>sum(irate(oc_ingressgateway_http_responses_total{namespace="\$namespace",Status=~"2.*",container="ingressgateway-prov",app_kubernetes_io_instance="\$Deployment"}[5m]))/sum(irate(oc_ingressgateway_http_requests_total{namespace="\$namespace",container="ingressgateway-prov",app_kubernetes_io_instance="\$Deployment"}[5m]))*100</code>

Table 5-2 Provisioning Request Rate(ingressgateway-prov)

Field	Details
Description	Measures the TPS of the provisioning traffic for each method wise and pod wise.
Expression	<code>sum(irate(oc_ingressgateway_http_requests_total{namespace="\$namespace",container="ingressgateway-prov",app_kubernetes_io_instance="\$Deployment"}[5m]))sum(irate(oc_ingressgateway_http_requests_total{namespace="\$namespace",container="ingressgateway-prov",app_kubernetes_io_instance="\$Deployment"}[5m]))by (Method)sum(irate(oc_ingressgateway_http_requests_total{namespace="\$namespace",container="ingressgateway-prov",app_kubernetes_io_instance="\$Deployment"}[5m]))by (pod)</code>

Table 5-3 Provisioning Response Breakdown

Field	Details
Sub KPI	Success Rate

Table 5-3 (Cont.) Provisioning Response Breakdown

Field	Details
Description	Measures the success percentage of provisioning requests that are on the ingressgateway-prov.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\sim\text{"2.*"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-4 Provisioning Response Breakdown

Field	Details
Sub KPI	Failure Rate
Description	Measures the failure rate of total requests that are on the ingressgateway-prov.
Expression	$\frac{(\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])) - \text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\sim\text{"2.*"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-5 Provisioning Response Breakdown

Field	Details
Sub KPI	408 response rate
Description	Measures the 408 response rate of the total requests that are on the ingressgateway-prov.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"408 REQUEST_TIMEOUT"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100 =$

Table 5-6 Provisioning Response Breakdown

Field	Details
Sub KPI	500 response rate
Description	Measures the 500 response rate of total requests that are on the ingressgateway-prov.

Table 5-6 (Cont.) Provisioning Response Breakdown

Field	Details
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"500 INTERNAL_SERVER_ERROR"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))/\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{*100}$

Table 5-7 Provisioning Response Breakdown

Field	Details
Sub KPI	503 response rate
Description	Measures the 503 response rate of the total requests that are on the ingressgateway-prov.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"503 SERVICE_UNAVAILABLE"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))/\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{*100}$

Table 5-8 Provisioning Response Breakdown

Field	Details
Sub KPI	404 response rate
Description	Measures the 404 response rate of the total requests that are on the ingressgateway-prov.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"404 NOT_FOUND"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))/\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{*100}$

Table 5-9 Provisioning Response Breakdown

Field	Details
Sub KPI	429 response rate
Description	Measures the 429 response rate of the total requests that are on the ingressgateway-prov.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"429 TOO_MANY_REQUESTS"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))/\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{*100}$

Table 5-10 Provisioning Response Breakdown

Field	Details
Sub KPI	400 response rate
Description	Measures the 400 response rate of the total requests that are on the ingressgateway-prov.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"400 BAD_REQUEST"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))/\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-prov"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{*100}$

Table 5-11 Signalling Success (2xx) Rate

Field	Details
Description	Measures the success rate percentage for signaling traffic.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"2.*"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))/\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{*100}$

Table 5-12 Signaling Request Rate(ingressgateway-sig)

Field	Details
Description	Measures the TPS of signaling traffic for each method wise and pod wise.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])) \text{ by (Method) } \text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])) \text{ by (pod)}}$

Table 5-13 Signaling Response Breakdown

Field	Details
Sub KPI	Success Rate
Description	Measures the success percentage of signaling requests that are on the ingressgateway-sig.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"2.*"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))/\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])))}{*100}$

Table 5-14 Signaling Response Breakdown

Field	Details
Sub KPI	Failure Rate
Description	Measures the failure percentage of the total requests that are on the ingressgateway-sig.
Expression	$\frac{(\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}])) - \text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"2.*"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-15 Signaling Response Breakdown

Field	Details
Sub KPI	408 response rate
Description	Measures the failure percentage of the total requests that are on the ingressgateway-sig.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"408 REQUEST_TIMEOUT"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-16 Signaling Response Breakdown

Field	Details
Sub KPI	500 response rate
Description	Measures the 500 response rate of the total requests that are on the ingressgateway-sig.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"500 INTERNAL_SERVER_ERROR"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-17 Signaling Response Breakdown

Field	Details
Sub KPI	503 response rate
Description	Measures the 503 response rate of the total requests that are on the ingressgateway-sig.

Table 5-17 (Cont.) Signaling Response Breakdown

Field	Details
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"503 SERVICE_UNAVAILABLE"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-18 Signaling Response Breakdown

Field	Details
Sub KPI	404 response rate
Description	Measures the 404 response rate of the total requests that are on the ingressgateway-sig.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"404 NOT_FOUND"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-19 Signaling Response Breakdown

Field	Details
Sub KPI	429 response rate
Description	Measures the 429 response rate of the total requests that are on the ingressgateway-sig.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"429 TOO_MANY_REQUESTS"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-20 Signaling Response Breakdown

Field	Details
Sub KPI	400 response rate
Description	Measures the 400 response rate of the total requests that are on the ingressgateway-sig.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{Status}=\text{"400 BAD_REQUEST"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{container}=\text{"ingressgateway-sig"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}\}[5\text{m}]))} * 100$

Table 5-21 UDR Incoming Traffic

Field	Details
Sub KPI	Sig Method Wise Request Rate
Description	Measures the request TPS that the backend pod (nudr-drservice) is currently handling method wise.
Expression	sum(irate(udr_nftype_specific_requests_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="nudr-drservice"}[5m])) by (Method)

Table 5-22 UDR Incoming Traffic

Field	Details
Sub KPI	Sig Method Wise Request Rate
Description	Measures the response TPS that the backend pod (nudr-drservice) is currently returning method wise.
Expression	sum(irate(udr_nftype_specific_requests_status_total{namespace="\$namespace",statusCode=~"2.*",app_kubernetes_io_instance="\$Deployment",container="nudr-drservice"}[5m])) by (Method)

Table 5-23 UDR Incoming Traffic

Field	Details
Sub KPI	Prov Method Wise Request Rate
Description	Measures the response TPS that the backend pod (nudr-drservice) is currently returning method wise.
Expression	sum(irate(udr_nftype_specific_requests_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="nudr-dr-provservice"}[5m])) by (Method)

Table 5-24 UDR Incoming Traffic

Field	Details
Sub KPI	Prov Method Wise Request Rate
Description	Measures the request TPS that the backend pod (nudr-drservice) is currently handling method wise.
Expression	sum(irate(udr_nftype_specific_requests_status_total{namespace="\$namespace",statusCode=~"2.*",app_kubernetes_io_instance="\$Deployment",container="nudr-dr-provservice"}[5m])) by (Method)

Table 5-25 IGW and DB Failure Rates

Field	Details
Sub KPI	Sig DB Failures
Description	Measures the database failures on the nudr-drservice for the signaling traffic.

Table 5-25 (Cont.) IGW and DB Failure Rates

Field	Details
Expression	$\frac{(\text{sum}(\text{irate}(\text{udr\_db\_operations\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"nudr-dr-service"}\}[5\text{m}])) - \text{sum}(\text{irate}(\text{udr\_db\_operations\_success\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"nudr-dr-service"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{udr\_db\_operations\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"nudr-dr-service"}\}[5\text{m}]))} * 100$

Table 5-26 IGW and DB Failure Rates

Field	Details
Sub KPI	Sig IGW Failures
Description	Measures the ingress gateway failure rates for the signaling traffic.
Expression	$\frac{(\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-sig"}\}[5\text{m}])) - \text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{status}=\text{"2.*"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-sig"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-sig"}\}[5\text{m}]))} * 100$

Table 5-27 IGW and DB Failure Rates

Field	Details
Sub KPI	Prov DB Failures
Description	Measures the database failures on the nudr-dr-service for the provisioning traffic.
Expression	$\frac{(\text{sum}(\text{irate}(\text{udr\_db\_operations\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"nudr-dr-prov-service"}\}[5\text{m}])) - \text{sum}(\text{irate}(\text{udr\_db\_operations\_success\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"nudr-dr-prov-service"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{udr\_db\_operations\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"nudr-dr-prov-service"}\}[5\text{m}]))} * 100$

Table 5-28 IGW and DB Failure Rates

Field	Details
Sub KPI	Prov IGW Failures
Description	Measures the ingress gateway failure rates for the provisioning traffic.
Expression	$\frac{(\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[5\text{m}])) - \text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_responses\_total}\{\text{namespace}=\text{"$namespace"}, \text{status}=\text{"2.*"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[5\text{m}]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_http\_requests\_total}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[5\text{m}]))} * 100$

Table 5-29 DB Operations Rate

Field	Details
Sub KPI	Total Operations Rate
Description	Measures the total number of database operations occurred from the backend microservices.
Expression	sum(irate(udr_db_operations_total{namespace="\$namespace"}[5m]))

Table 5-30 DB Operations Rate

Field	Details
Sub KPI	Success Rate
Description	Measures the total number of successful database operations occurred from the backend microservices.
Expression	sum(irate(udr_db_operations_success_total{namespace="\$namespace"}[5m]))

Table 5-31 DB Operations Rate

Field	Details
Sub KPI	Failure Rate
Description	Measures the total number of failed database operations occurred from backend microservices.
Expression	sum(irate(udr_db_operations_total{namespace="\$namespace"}[5m]))- sum(irate(udr_db_operations_success_total{namespace="\$namespace"}[5m]))

Table 5-32 IGW Latencies(Percentile)

Field	Details
Sub KPI	Backend latency Sig
Description	Measures the signaling traffic backend pod latencies.
Expression	histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_backend_invocation_latency_seconds_bucket{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])))

Table 5-33 IGW Latencies(Percentile)

Field	Details
Sub KPI	GW Request latency Sig
Description	Measures the gateway latency which are on the ingressgateway-sig.
Expression	histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_request_processing_latency_seconds_bucket{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])))

Table 5-34 IGW Latencies(Percentile)

Field	Details
Sub KPI	GW Response latency Sig
Description	Measures the gateway response latency for signaling traffic.
Expression	<code> histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_response_processing_latency_seconds_bucket{namesp ace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingre ssgateway-sig"}[2m]))) </code>

Table 5-35 IGW Latencies(Percentile)

Field	Details
Sub KPI	GW total latency Sig
Description	Measures the total latency (including request, response, and processing) for each of the request that are on the ingressgateway-sig.
Expression	<code> (histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_backend_invocation_latency_seconds_bucket{namespac e="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingress gateway-sig"}[2m]))) + histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_request_processing_latency_seconds_bucket{namespac e="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingress gateway-sig"}[2m]))) + histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_response_processing_latency_seconds_bucket{namesp ace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingre ssgateway-sig"}[2m]))) </code>

Table 5-36 IGW Latencies(Percentile)

Field	Details
Sub KPI	Backend latency Prov
Description	Measures the provisioning traffic backend pod latencies.
Expression	<code> histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_backend_invocation_latency_seconds_bucket{namespac e="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingress gateway-prov"}[2m]))) </code>

Table 5-37 IGW Latencies(Percentile)

Field	Details
Sub KPI	GW Request latency Prov
Description	Measures the gateway latency that are on the ingressgateway-prov.
Expression	<code> histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_request_processing_latency_seconds_bucket{namespac e="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingress gateway-prov"}[2m]))) </code>

Table 5-38 IGW Latencies(Percentile)

Field	Details
Sub KPI	GW Response latency Prov
Description	Measures the gateway response latency for the provisioning traffic.
Expression	<code> histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_response_processing_latency_seconds_bucket{namesp ace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingre ssgateway-prov"}[2m]))) </code>

Table 5-39 IGW Latencies(Percentile)

Field	Details
Sub KPI	GW total latency Prov
Description	Measures the total latency (including request, response, and processing) for each of the request that are on the ingressgateway-prov.
Expression	<code> (histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_backend_invocation_latency_seconds_bucket{namespac e="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingress gateway-prov"}[2m]))) + histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_request_processing_latency_seconds_bucket{namespac e="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingress gateway-prov"}[2m]))) + histogram_quantile(\${percentile} / 100, sum by(le) (rate(oc_ingressgateway_response_processing_latency_seconds_bucket{namesp ace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingre ssgateway-prov"}[2m]))) </code>

Table 5-40 DB and DR-Service Latency [Percentile]

Field	Details
Sub KPI	udr_sig_db_latency
Description	Measures the signaling traffic database latency.
Expression	<code> histogram_quantile(\${percentile} / 100, sum by(le) (rate(udr_db_processing_time_seconds_bucket{namespace="\$namespace",app_k ubernetes_io_instance="\$Deployment",container="nudr-drservice"}[5m]))) </code>

Table 5-41 DB and DR-Service Latency [Percentile]

Field	Details
Sub KPI	udr_sig_request_latency
Description	Measures the signaling traffic request latency.
Expression	<code> histogram_quantile(\${percentile} / 100, sum by(le) (rate(udr_request_processing_time_seconds_bucket{namespace="\$namespace",a pp_kubernetes_io_instance="\$Deployment",container="nudr-drservice"}[5m]))) </code>

Table 5-42 DB and DR-Service Latency [Percentile]

Field	Details
Sub KPI	udr_prov_db_latency

Table 5-42 (Cont.) DB and DR-Service Latency [Percentile]

Field	Details
Description	Measures the provisioning traffic database latency.
Expression	<code> histogram_quantile({percentile} / 100, sum by(le) (rate(udr_db_processing_time_seconds_bucket{namespace="\$namespace",app_k ubernetes_io_instance="\$Deployment",container="nudr-dr-provservice"}[5m]))) </code>

Table 5-43 DB and DR-Service Latency [Percentile]

Field	Details
Sub KPI	udr_prov_request_latency
Description	Measures the provisioning traffic request latency.
Expression	<code> histogram_quantile({percentile} / 100, sum by(le) (rate(udr_request_processing_time_seconds_bucket{namespace="\$namespace",a pp_kubernetes_io_instance="\$Deployment",container="nudr-dr-provservice"} [5m]))) </code>

Table 5-44 IGW Latencies [Average]

Field	Details
Sub KPI	Backend latency Sig
Description	Measures the average latency for the backend services for signaling traffic.
Expression	<code> sum(irate(oc_ingressgateway_backend_invocation_latency_seconds_sum{namesp ace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingre ssgateway-sig"}[2m])) / sum(irate(oc_ingressgateway_backend_invocation_latency_seconds_count{name space="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingr essgateway-sig"}[2m])) </code>

Table 5-45 IGW Latencies [Average]

Field	Details
Sub KPI	GW Request latency Sig
Description	Measures the average request latency for signaling traffic.
Expression	<code> (sum(irate(oc_ingressgateway_request_processing_latency_seconds_sum{name space="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingr essgateway-sig"}[2m])) / sum(irate(oc_ingressgateway_request_processing_latency_seconds_count{name space="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingr essgateway-sig"}[2m]))) </code>

Table 5-46 IGW Latencies [Average]

Field	Details
Sub KPI	GW Response latency Sig
Description	Measures the average response latency for signaling traffic.

Table 5-46 (Cont.) IGW Latencies [Average]

Field	Details
Expression	<code>(sum(irate(oc_ingressgateway_response_processing_latency_seconds_sum{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])) / sum(irate(oc_ingressgateway_response_processing_latency_seconds_count{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])))</code>

Table 5-47 IGW Latencies [Average]

Field	Details
Sub KPI	GW total latency Sig
Description	Measures the average total latency that includes request, response, and processing for signaling traffic.
Expression	<code>(sum(irate(oc_ingressgateway_backend_invocation_latency_seconds_sum{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])) / sum(irate(oc_ingressgateway_backend_invocation_latency_seconds_count{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m]))) + (sum(irate(oc_ingressgateway_request_processing_latency_seconds_sum{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])) / sum(irate(oc_ingressgateway_request_processing_latency_seconds_count{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m]))) + (sum(irate(oc_ingressgateway_response_processing_latency_seconds_sum{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])) / sum(irate(oc_ingressgateway_response_processing_latency_seconds_count{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-sig"}[2m])))</code>

Table 5-48 IGW Latencies [Average]

Field	Details
Sub KPI	Backend latency Prov
Description	Measures the average latency for backend services for provisioning traffic.
Expression	<code>(sum(irate(oc_ingressgateway_backend_invocation_latency_seconds_sum{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-prov"}[2m])) / sum(irate(oc_ingressgateway_backend_invocation_latency_seconds_count{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment",container="ingressgateway-prov"}[2m])))</code>

Table 5-49 IGW Latencies [Average]

Field	Details
Sub KPI	GW Request latency Prov
Description	Measures the average request latency for provisioning traffic.

Table 5-49 (Cont.) IGW Latencies [Average]

Field	Details
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_request\_processing\_latency\_seconds\_sum}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_request\_processing\_latency\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))})$

Table 5-50 IGW Latencies [Average]

Field	Details
Sub KPI	GW Response latency Prov
Description	Measures the average response latency for provisioning traffic.
Expression	$\frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_response\_processing\_latency\_seconds\_sum}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_response\_processing\_latency\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))})$

Table 5-51 IGW Latencies [Average]

Field	Details
Sub KPI	GW total latency Prov
Description	Measures the average total latency that includes request, response, and processing for provisioning traffic.
Expression	$\begin{aligned} & \frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_backend\_invocation\_latency\_seconds\_sum}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_backend\_invocation\_latency\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))})} + \\ & \frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_request\_processing\_latency\_seconds\_sum}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_request\_processing\_latency\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))})} + \\ & \frac{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_response\_processing\_latency\_seconds\_sum}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))}{\text{sum}(\text{irate}(\text{oc\_ingressgateway\_response\_processing\_latency\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{app_kubernetes_io_instance}=\text{"$Deployment"}, \text{container}=\text{"ingressgateway-prov"}\}[2m]))})} \end{aligned}$

Table 5-52 Error Logs Indicator

Field	Details
Sub KPI	Microservice wise error logs indicator

Table 5-52 (Cont.) Error Logs Indicator

Field	Details
Description	Measures the error logs that are printing for the respective microservices. This helps in debugging purpose and understand the error on the microservice pod level.
Expression	sum by (microservice) (increase(log4j2_events_total{namespace="\$namespace",level="error",app_kubernetes_io_instance="\$Deployment"}[3m]))

Table 5-53 CPU Usage Across Namespace (per pod)

Field	Details
Sub KPI	POD Wise CPU Utilization
Description	Measures the CPU resource usage for all the pods that are deployed within the namespace.
Expression	irate(cgroup_cpu_nanoseconds{namespace="\$namespace"}[5m])

Table 5-54 Memory Usage Across Namespace (per pod)

Field	Details
Sub KPI	POD Wise Memory Utilization
Description	Measures the memory resource usage for all the pods that are deployed with in the namespace.
Expression	sum(cgroup_memory_bytes{namespace="\$namespace"}) by (pod)

Table 5-55 Overload Level

Field	Details
Sub KPI	Overload Level
Description	Measures the overall load level for the backend pod that is handling the signaling or provisioning traffic.
Expression	load_level{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"} > 0

Table 5-56 Overload level resource wise

Field	Details
Sub KPI	Overload level resource wise
Description	Measures the overload level by resource. Example as follows: - CPU is at L1 - failure count is L2
Expression	max by (type, service) (service_resource_overload_level{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}) > 0

Table 5-57 Overload Stress Resource Wise

Field	Details
Sub KPI	Overload stress resource wise
Description	Measures the overload stress by resource. Example as follows: - CPU is at 40 - failure count is at 8000
Expression	max by (type, service) (service_resource_stress{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}) > 0

Table 5-58 IGW Pod Congested State

Field	Details
Sub KPI	IGW Pod Congested State
Description	Measures the of ingress gateway pod congested state level. It may be at NORMAL, DOC or CONGESTED state.
Expression	sum by (pod) (oc_ingressgateway_pod_congestion_state{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"})

Table 5-59 IGW Congestion State Resource Wise

Field	Details
Sub KPI	IGW Congestion state resource wise
Description	Measures the congested level by resource. Example as follows: - CPU is at DOC - pending count is CONGESTED
Expression	max by (type, pod) (oc_ingressgateway_pod_resource_state{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"})

Table 5-60 POD Wise Congested Resource Stress

Field	Details
Sub KPI	POD wise congested resource stress
Description	Measures the overload stress by resource. Example as follows: - CPU is at 40 - pending count is at 8000
Expression	max by (type, pod) (oc_ingressgateway_pod_resource_stress{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}) > 0

Table 5-61 IGW Total Connections Rejected

Field	Details
Sub KPI	IGW Total Connections Rejected

Table 5-61 (Cont.) IGW Total Connections Rejected

Field	Details
Description	Measures the total number of connections that got rejected at ingress gateway level because of the CONGESTED level.
Expression	<code>sum(irate(oc_ingressgateway_incoming_pod_connections_rejected_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])) by (pod)</code>

Table 5-62 NRF Client Operative State

Field	Details
Sub KPI	NRF Client NF Status
Description	Measures the NRF Client operative state. 0 - NRF is unavailable or unhealthy
Expression	<code>nrfclient_nrf_operative_status{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}</code>

Table 5-63 NRF Client NF Status

Field	Details
Sub KPI	NRF Client NF Status
Description	Measures the NRF Client NF Status. 1 - DEREGISTERED 2 - SUSPENDED 3 - UNDISCOVERABLE 4 - UNKNOWN
Expression	<code>nrfclient_nf_status_with_nrf{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}</code>

Table 5-64 kmeps per group(cgroup)

Field	Details
Sub KPI	Ingressgateway-sig
Description	Measures the kmeps of ingressgateway-sig service.
Expression	<code>(sum(rate(http_server_requests_seconds_count{namespace="\$namespace",pod=~".*ingressgateway-sig.*"}[2m]))*2)/sum(rate(cgroup_cpu_nanoseconds{namespace="\$namespace",pod=~".*ingressgateway-sig.*"}[2m])/1E9)</code>

Table 5-65 kmeps per group(cgroup)

Field	Details
Sub KPI	Egressgateway
Description	Measures the kmeps of egressgateway service.

Table 5-65 (Cont.) kmps per group(cgroup)

Field	Details
Expression	$\frac{(\text{sum}(\text{rate}(\text{http\_server\_requests\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*egressgateway.*"}\}[2\text{m}]))*2)}{\text{sum}(\text{rate}(\text{cgroup_cpu_nanoseconds}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*egressgateway.*"}\}[2\text{m}])/1\text{E9})}$

Table 5-66 kmps per group(cgroup)

Field	Details
Sub KPI	DR Service
Description	Measures the kmps of nudr-dr service.
Expression	$\frac{(\text{sum}(\text{rate}(\text{http\_server\_requests\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*nudur-dr-service.*"}\}[2\text{m}]))}{\text{sum}(\text{rate}(\text{cgroup_cpu_nanoseconds}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*nudur-dr-service.*"}\}[2\text{m}])/1\text{E9})}$

Table 5-67 kmps per group(cgroup)

Field	Details
Sub KPI	Ingressgateway-prov
Description	Measures the kmps of ingressgateway-prov service.
Expression	$\frac{(\text{sum}(\text{rate}(\text{http\_server\_requests\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*ingressgateway-prov.*"}\}[2\text{m}]))*2)}{\text{sum}(\text{rate}(\text{cgroup_cpu_nanoseconds}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*ingressgateway-prov.*"}\}[2\text{m}])/1\text{E9})}$

Table 5-68 kmps per group(cgroup)

Field	Details
Sub KPI	DR-Prov Service
Description	Measures the kmps of nudr-dr-prov service.
Expression	$\frac{(\text{sum}(\text{rate}(\text{http\_server\_requests\_seconds\_count}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*nudur-dr-prov-service.*"}\}[2\text{m}]))}{\text{sum}(\text{rate}(\text{cgroup_cpu_nanoseconds}\{\text{namespace}=\text{"$namespace"}, \text{pod}=\sim\text{"}.\text{*nudur-dr-prov-service.*"}\}[2\text{m}])/1\text{E9})}$

5.1.1 UDR and EIR KPIs

Following table provides UDR and EIR KPIs.

Table 5-69 Diameter Gateway

Field	Details
Sub KPI	Diameter GW Success Rate
Description	Measures the total success TPS rate of the diameter gateway service.

Table 5-69 (Cont.) Diameter Gateway

Field	Details
Expression	<code>sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",responseCode="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))/sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])) *100</code>

Table 5-70 Diameter Gateway

Field	Details
Sub KPI	Diameter GW Failure Rate
Description	Measures the total failure TPS rate of the diameter gateway service.
Expression	<code>sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",responseCode!="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))/sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])) *100</code>

Table 5-71 Diameter Gateway Traffic Rate

Field	Details
Sub KPI	Total Request Rate
Description	Measures the total TPS of the diameter traffic that is on the diameter gateway.
Expression	<code>sum(irate(ocudr_diam_request_network_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))</code>

Table 5-72 Diameter Gateway Traffic Rate

Field	Details
Sub KPI	Success Rate
Description	Measures the total successful TPS of the diameter traffic that is on the diameter gateway.
Expression	<code>sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",responseCode="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))</code>

Table 5-73 Diameter Gateway Traffic Rate

Field	Details
Sub KPI	Failure Rate
Description	Measures the total failed TPS of the diameter traffic that is on the diameter gateway.
Expression	<code>sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",responseCode!="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))</code>

Table 5-74 Diameter Gateway Traffic Rate

Field	Details
Sub KPI	Message Type Wise TPS
Description	Measures the total TPS of the diameter traffic that is on the diameter gateway by message type.
Expression	sum(irate(ocudr_diam_request_network_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])) by (msgType)

Table 5-75 Diameter Gateway Traffic Rate

Field	Details
Sub KPI	Message Type Wise Success Rate
Description	Measures the total successful TPS of the diameter traffic that is on the diameter gateway by message type.
Expression	sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",responseCode="2001",app_kubernetes_io_instance="\$Deployment"}[5m])) by (msgType)

Table 5-76 Diameter Gateway Traffic Rate

Field	Details
Sub KPI	Message Type Wise Failure Rate
Description	Measures the total failure TPS of the diameter traffic that is on the diameter gateway by message type.
Expression	sum(irate(ocudr_diam_response_network_total{namespace="\$namespace",responseCode!="2001",app_kubernetes_io_instance="\$Deployment"}[5m])) by (msgType)

Table 5-77 Diameter Backend

Field	Details
Sub KPI	Diameter Backend Success Rate
Description	Measures the total successful TPS rate of the diameter proxy service.
Expression	sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",Result Code="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))/sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])) *100

Table 5-78 Diameter Backend

Field	Details
Sub KPI	Diameter Backend Failure Rate
Description	Measures the total failure TPS rate of the diameter proxy service.
Expression	sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",Result Code!="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))/sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])) *100

Table 5-79 Diameter Backend Traffic Rate

Field	Details
Sub KPI	Total Request Rate
Description	Measures the total TPS of the diameter traffic that is on diameter proxy.
Expression	sum(irate(nudr_diameterproxy_requests_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-80 Diameter Backend Traffic Rate

Field	Details
Sub KPI	Success Rate
Description	Measures the total success TPS of the diameter traffic that is on diameter proxy.
Expression	sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",Result Code="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-81 Diameter Backend Traffic Rate

Field	Details
Sub KPI	Failure Rate
Description	Measures the total failure TPS of the diameter traffic that is on diameter proxy.
Expression	sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",Result Code!="2001",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-82 Diameter Backend Traffic Rate

Field	Details
Sub KPI	Message Type Wise Total TPS
Description	Measures the total TPS of the diameter traffic that is on diameter proxy by message type.
Expression	sum(irate(nudr_diameterproxy_requests_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])) by (MessageType)

Table 5-83 Diameter Backend Traffic Rate

Field	Details
Sub KPI	Message Type Wise Success Rate
Description	Measures the total success TPS of the diameter traffic that is on diameter proxy by message type.
Expression	sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",Result Code="2001",app_kubernetes_io_instance="\$Deployment"}[5m])) by (MessageType)

Table 5-84 Diameter Backend Traffic Rate

Field	Details
Sub KPI	Message Type Wise Failure Rate
Description	Measures the total failure TPS of the diameter traffic that is on diameter proxy by message type.
Expression	sum(irate(nudr_diameterproxy_responses_total{namespace="\$namespace",Result Code!="2001",app_kubernetes_io_instance="\$Deployment"}[5m])) by (MessageType)

Table 5-85 DB and DR-Service Latency [Percentile]

Field	Details
Sub KPI	diameter_db_latency
Description	Measures the diameter traffic database latency.
Expression	histogram_quantile({percentile} / 100, sum by(le) (rate(diameter_db_processing_time_seconds_bucket{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])))

Table 5-86 DB and DR-Service Latency [Percentile]

Field	Details
Sub KPI	diameter_req_latency
Description	Measures the diameter traffic request latency.
Expression	histogram_quantile({percentile} / 100, sum by(le) (rate(diameter_request_processing_time_seconds_bucket{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])))

Table 5-87 DB and DR-Service Latency [Percentile]

Field	Details
Sub KPI	diameter_backend_latency
Description	Measures the diameter traffic backend request latency.
Expression	histogram_quantile({percentile} / 100, sum by(le) (rate(diamgateway_backend_processing_time_seconds_bucket{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m])))

Table 5-88 DGW Congestion State

Field	Details
Sub KPI	DGW Congestion State
Description	Measures the congested state of Diameter Gateway pod.
Expression	sum by (pod) (ocudr_pod_congestion_state{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"})

Table 5-89 DGW Congestion State Resource Wise

Field	Details
Sub KPI	DGW congestion state resource wise
Description	Measures the diameter gateway congested level by resource. Example as follows: - CPU is at DOC - pending count is CONGESTED
Expression	max by (type, pod) (ocudr_pod_resource_congestion_state{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"})

Table 5-90 DGW Congestion Resource Stress

Field	Details
Sub KPI	DGW congestion resource stress
Description	Measures the diameter gateway overload stress by resource. Example as follows: - CPU is at 40 - pending count is at 8000
Expression	max by (type, pod) (ocudr_pod_resource_congestion_state{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}) > 0

Table 5-91 DGW Total Messages Rejected

Field	Details
Sub KPI	DGW Total Messages Rejected
Description	Measures the total number of connections that is rejected at diameter gateway because of CONGESTED level.
Expression	ocudr_diam_congestion_message_reject_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}

5.2 UDR KPIs

Following table provides the UDR KPIs.

5.2.1 UDR Notifications

Following table provides the UDR Notifications KPIs.

Table 5-92 Notifications Rate (NOTIFY SERVICE)

Field	Details
Sub KPI	Notifications sent
Description	Measures the total notifications rate that are sent from the notify-service.
Expression	sum(irate(nudr_notif_notifications_sent_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-93 Notifications Rate (NOTIFY SERVICE)

Field	Details
Sub KPI	Total ack received
Description	Measures the TPS of the received notifications to notify-service.
Expression	sum(irate(nudr_notif_notifications_ack_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-94 Notifications Rate (NOTIFY SERVICE)

Field	Details
Sub KPI	2xx ack received
Description	Measures the TPS of 2xx received notifications to notify-service.
Expression	sum(irate(nudr_notif_notifications_ack_2xx_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-95 Notifications Rate (NOTIFY SERVICE)

Field	Details
Sub KPI	4xx ack received
Description	Measures the TPS of 4xx received notifications to notify-service.
Expression	sum(irate(nudr_notif_notifications_ack_4xx_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-96 Notifications Rate (NOTIFY SERVICE)

Field	Details
Sub KPI	5xx ack received
Description	Measures the TPS of 5xx received notifications to notify-service.
Expression	sum(irate(nudr_notif_notifications_ack_5xx_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-97 Notification Rate (DR SERVICE/DIAM PROXY)

Field	Details
Sub KPI	Notifications Sent DR
Description	Measures the total notifications that are sent to notify-service from the dr-service.
Expression	sum(irate(nudr_dr_notif_signal_sent_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-98 Notification Rate (DR SERVICE/DIAM PROXY)

Field	Details
Sub KPI	UPDATE Notifications Failure Rate DR
Description	Measures the total failed update notifications that are sent to notify-service from the dr-service.

Table 5-98 (Cont.) Notification Rate (DR SERVICE/DIAM PROXY)

Field	Details
Expression	sum(irate(nudr_dr_notif_signal_notification_update_failed_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-99 Notification Rate (DR SERVICE/DIAM PROXY)

Field	Details
Sub KPI	UPDATE Notifications Success Rate
Description	Measures the total successfully update notifications that are sent to notify-service from the dr-service.
Expression	sum(irate(nudr_dr_notif_signal_notification_update_success_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-100 Notification Rate (DR SERVICE/DIAM PROXY)

Field	Details
Sub KPI	DELETE Notifications Success Rate DR
Description	Measures the total delete notifications that are successfully sent to notify-service from the dr-service.
Expression	sum(irate(nudr_dr_notif_signal_notification_delete_success_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-101 Notification Rate (DR SERVICE/DIAM PROXY)

Field	Details
Sub KPI	DELETE Notifications Failure Rate DR
Description	Measures the total failed delete notifications to notify-service from the dr-service.
Expression	sum(irate(nudr_dr_notif_signal_notification_delete_failed_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-102 Retry and New Notifications

Field	Details
Sub KPI	Retry Notification
Description	Measures the retry notifications that are retried.
Expression	sum(irate(nudr_notif_notifications_marked_for_retry_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-103 Retry and New Notifications

Field	Details
Sub KPI	New Notifications
Description	Measures the new notifications that are sent from notify service.

Table 5-103 (Cont.) Retry and New Notifications

Field	Details
Expression	sum(irate(nudr_notif_notifications_sent_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))- sum(irate(nudr_notif_notifications_marked_for_retry_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

5.2.2 On-Demand Migration

Table 5-104 On-Demand Migration

Field	Details
Sub KPI	No of requests received
Description	Measures the total number of requests received to migrate data using on-demand migration.
Expression	sum(irate(nudr_ondemand_migration_requests_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-105 On-Demand Migration

Field	Details
Sub KPI	No of requests sent to 4G UDR
Description	Measures the total requests sent to 4G UDR.
Expression	sum(irate(nudr_ondemand_migration_requests_read_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-106 On-Demand Migration

Field	Details
Sub KPI	No of requests migrated to 5G UDR
Description	Measures the total requests sent to 5G UDR.
Expression	sum(irate(nudr_ondemand_migration_requests_processed_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-107 On-Demand Migration

Field	Details
Sub KPI	No of subscribers not exist in 4G UDR
Description	Measures the total number of subscribers that do not exist in 4G UDR.
Expression	sum(irate(nudr_ondemand_migration_subscriber_not_exist_source_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-108 On-Demand Migration

Field	Details
Sub KPI	No of subscribers not exist in 4G UDR
Description	Measures the total number of deleted subscribers that do not exist in 4G UDR.
Expression	sum(irate(nudr_ondemand_migration_delete_subscriber_source_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-109 On-Demand Migration

Field	Details
Sub KPI	No of subscribers failed while retrieving from 4G UDR
Description	Measures the total number of subscribers failed while retrieving from 4G UDR.
Expression	sum(irate(nudr_ondemand_migration_requests_read_status_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

Table 5-110 On-Demand Migration

Field	Details
Sub KPI	No of subscribers failed during migration to 5G UDR
Description	Measures the total number of subscribers failed during migration to 5G UDR.
Expression	sum(irate(nudr_ondemand_migration_requests_processed_status_total{namespace="\$namespace",app_kubernetes_io_instance="\$Deployment"}[5m]))

5.3 SLF KPIs

Following section provides the SLF KPIs.

5.3.1 Ingress Gateway Pod Protection Using Rate Limiting

Following table provides Ingress Gateway Pod Protection Using Rate Limiting KPIs

Table 5-111 Congestion Level

Field	Details
Sub KPI	Congestion Level
Description	Measures the congestion level per pod at particular instance.
Expression	sum(oc_ingressgateway_congestion_system_state{namespace="\$namespace",container="ingressgateway-sig"}) by (pod)

Table 5-112 Allowed/Rejected Requests

Field	Details
Sub KPI	Allowed requests
Description	Measures the total allowed requests across all the pods.

Table 5-112 (Cont.) Allowed/Rejected Requests

Field	Details
Expression	sum(rate(oc_ingressgateway_http_request_ratelimit_values_total{Allowed="true", namespace="\$namespace", container="ingressgateway-sig"}[2m])) by (Id)

Table 5-113 Allowed/Rejected Requests

Field	Details
Sub KPI	Rejected requests
Description	Measures the total rejected requests across all the pods.
Expression	sum(rate(oc_ingressgateway_http_request_ratelimit_values_total{Allowed="false", namespace="\$namespace", container="ingressgateway-sig"}[2m])) by (Id)

Table 5-114 Allow/Reject Requests per Pod

Field	Details
Sub KPI	Allow/Reject requests per pod
Description	Measures the CONTINUE or REJECT requests per pod along with the congestion level.
Expression	sum(rate(oc_ingressgateway_http_request_ratelimit_denied_count_total{namespace="\$namespace", container="ingressgateway-sig"}[2m])) by (Action,pod)

Table 5-115 Overall CPU Usage

Field	Details
Sub KPI	Overall CPU Usage
Description	Measures the overall CPU usage for the ingressgateway-sig microservice.
Expression	sum(rate(oc_ingressgateway_congestion_cpu_percentage_bucket{namespace="\$namespace", container="ingressgateway-sig"}[2m])) by (le)

6

Metrics

This section provides information about the Metrics for Oracle Communications Unified Data Repository (OCUDR).

Note

- A sample Grafana json is available in the UDR Customer Documentation on Oracle Help Center. You can download and import the file to your repository. <http://docs.oracle.com/en/industries/communications/cloud-native-core/release21.html>

Table 6-1 Metric Type

Metric Type	Description
Counter	Represents the total number of occurrences of an event or traffic, such as measuring the total amount of traffic received and transmitted by UDR. Prometheus automatically appends the words, such as total to all the counter type metrics that are displayed on the Prometheus dashboard.
Gauge	Represents a single numerical value that changes randomly. This metric type is used to measure various parameters, such as UDR load values, memory usage, and so on.
Histogram	Represents the number of occurrences that fall into predefined intervals, or buckets, such as those used for measuring response times. Prometheus automatically appends the words, such as bucket, count, and sum to all the histogram metrics that are displayed on the Prometheus dashboard.

6.1 Common Metrics

Table 6-2 security_cert_x509_expiration_seconds

Field	Details
Metric Name	security_cert_x509_expiration_seconds
Description	Indicates the time to certificate expiry in epoch seconds.
Type	Histogram
Dimensions	serialNumber

Table 6-3 oc_oauth_certificate_loadStatus

Field	Details
Metric Name	oc_oauth_certificate_loadStatus
Description	Gauge metric is used to peg whether the given certificate can be loaded from secret or not. If it can be loaded, then 0 is pegged, otherwise 1 is pegged. This metric is further used for raising alarms when certificate cannot be loaded.
Type	- id - certificateName - secretName
Dimensions	NA

Table 6-4 oc_oauth_certificate_expiryStatus

Field	Details
Metric Name	oc_oauth_certificate_expiryStatus
Description	Gauge metric is used to peg expiry date of the certificate. This metric is further used for raising alarms if certificate expires within 30 days or 7 days.
Type	- id - certificateName - secretName
Dimensions	NA

6.2 UDR Gateways Metrics

This section provides details about the Gateway metrics of UDR.

6.2.1 Ingress Gateway Metrics

The following metrics are applicable to Ingress API Gateway.

Table 6-5 Total Ingress Requests

Field	Details
Metric Name	oc_ingressgateway_http_requests_total{Method="POST/PUT/GET/DELETE/PATCH", Scheme="HTTP/HTTPS",container="ingressgateway-sig"}
Description	This metric is pegged as soon as the request reaches the Ingress gateway in the first custom filter of the application.
Type	Counter
Dimensions	Method NFType NFServiceType Host HttpVersion Scheme

Table 6-6 Total Ingress Responses

Field	Details
Metric Name	oc_ingressgateway_http_responses_total{Method="POST/PUT/GET/DELETE/PATCH", Status=~"2.* 4.* 5.*",container="ingressgateway-sig"}
Description	Total number of HTTP responses sent from Ingress Gateway.
Type	Counter
Dimensions	Status Method NFType NFServiceType Host HttpVersion Scheme

Table 6-7 Avg Latency

Field	Details
Metric Name	oc_ingressgateway_request_latency_seconds{container="ingressgateway-sig"}
Description	Time (in microseconds) to process an ingress request. Measured when the request is received and the response is sent.
Type	Timer
Dimensions	quantile InstanceIdentifier

Table 6-8 Connection Failures Ingress Gateway

Field	Details
Metric Name	oc_ingressgateway_connection_failure_total{container="ingressgateway-sig"}
Description	Metric to capture the connection failures when connect to the destination service fails. Here in case of Ingress Gateway, the destination service is a backend microservice of the NF. And TLS connection failure metrics when connecting to ingress.
Type	Counter
Dimensions	Host Port InstanceIdentifier error_reason

Table 6-9 Ingress Gateway Request Processing Latency

Field	Details
Metric Name	oc_ingressgateway_request_processing_latency_seconds{container="ingressgateway-sig"}
Description	Metric to capture the amount of time taken for processing of the request only within Ingress Gateway.
Type	Timer

Table 6-9 (Cont.) Ingress Gateway Request Processing Latency

Field	Details
Dimensions	quantile InstanceIdentifier

Table 6-10 User-Agent Successful validation

Field	Details
Metric Name	oc_ingressgateway_user_agent_consumerNfType_success_count
Description	This metric is pegged for every incoming request when the feature is enabled and validation is successful.
Type	Counter
Dimensions	id - id of the route consumerNfType - consumer NF type obtained from the User-Agent header in the incoming request

Table 6-11 User-Agent Failure validation

Field	Details
Metric Name	oc_ingressgateway_user_agent_consumerNfType_failure_count
Description	For every incoming request when the feature is enabled and user agent validation fails because of mismatch in the value configured and the value present in the incoming request.
Type	Counter
Dimensions	id - id of the route consumerNfType - consumer NF Type obtained from the User-Agent header in the incoming request.

Table 6-12 User-Agent Invalid Config

Field	Details
Metric Name	oc_ingressgateway_useragent_invalid_config
Description	If the user agent configuration is invalid that is if the consumerNFType list is empty, then it is applicable only for HELM based configuration.
Type	Gauge
Dimensions	NA

Table 6-13 User-Agent Invalid Config

Field	Details
Metric Name	oc_ingressgateway_sbimer_timezone_mismatch
Description	This metric is pegged when sbiTimerTimezone is set to ANY and the timezone is not specified in the header, then the above metric is pegged in ingress and egress gateway.
Type	Gauge
Dimensions	NA

Table 6-14 Ingress Gateway Pod Congestion State

Field	Details
Metric Name	oc_ingressgateway_pod_congestion_state{namespace='\$namespace'}
Description	Tracks the overall congestion state per ingress gateway pod.
Type	Counter
Dimensions	NA

Table 6-15 Ingress Gateway Pod Resource State

Field	Details
Metric Name	oc_ingressgateway_pod_resource_state{type="CPU/PENDING_REQUEST",namespace='\$namespace'}
Description	Tracks the resource congestion state per ingress gateway pod.
Type	Counter
Dimensions	NA

Table 6-16 Ingress Gateway Pod Connections Rejected

Field	Details
Metric Name	oc_ingressgateway_incoming_pod_connections_rejected_total{namespace="\$namespace"}
Description	Tracks the resource usage per Ingress Gateway pod.
Type	Counter
Dimensions	NA

Table 6-17 Ingress Gateway Pod Incoming Connection Rejections

Field	Details
Metric Name	oc_ingressgateway_incoming_pod_connections_rejected_total{namespace="\$namespace"}
Description	Tracks the incoming rejected connections.
Type	Counter
Dimensions	NA

Table 6-18 Total Ingress Requests

Field	Details
Metric Name	oc_ingressgateway_http_requests_total{Method="POST/PUT/GET/DELETE/PATCH",Scheme="HTTP/HTTPS",container="ingressgateway-prov"}
Description	Total number of requests received by ingress gateway.
Type	Counter
Dimensions	Method NFType NFServiceType Host HttpVersion Scheme

Table 6-19 Total Ingress Responses

Field	Details
Metric Name	oc_ingressgateway_http_responses_total{Method="POST/PUT/GET/DELETE/PATCH", Status=~"2.* 4.* 5.*",container="ingressgateway-prov"}
Description	Total number of requests received by ingress gateway.
Type	Counter
Dimensions	Status Method NFType NFServiceType Host HttpVersion Scheme

Table 6-20 Avg Latency

Field	Details
Metric Name	oc_ingressgateway_request_latency_seconds{container="ingressgateway-prov"}
Description	Time (in microseconds) to process an ingress request. Measured when the request is received and the response is sent
Type	Timer
Dimensions	quantile InstanceIdentifier

Table 6-21 Connection Failures Ingress Gateway

Field	Details
Metric Name	oc_ingressgateway_connection_failure_total{container="ingressgateway-prov"}
Description	Metric to capture the connection failures when the connect to the destination service fails. In case of Ingress Gateway, the destination service is a backend microservice of the NF. TLS connection failure metrics when connecting to ingress.
Type	Counter
Dimensions	Host Port InstanceIdentifier error_reason

Table 6-22 Ingress Gateway Request Processing Latency

Field	Details
Metric Name	oc_ingressgateway_request_processing_latency_seconds{container="ingressgateway-prov"}
Description	Metric to capture the amount of time taken for processing the request only within Ingress Gateway
Type	Timer

Table 6-22 (Cont.) Ingress Gateway Request Processing Latency

Field	Details
Dimensions	quantile InstanceIdentifier

Table 6-23 Ingress Gateway Pod Resource Stress

Field	Details
Metric Name	oc_ingressgateway_pod_resource_stress{type="CPU/PENDING_REQUEST",namespace='\$namespace'}
Description	Tracks the resource usage per Ingress Gateway pod.
Type	Counter
Dimensions	NA

Table 6-24 oc_ingressgateway_incoming_tls_connections

Field	Details
Metric Name	oc_ingressgateway_incoming_pod_connections_rejected_total{namespace="\$namespace"}
Description	Number of TLS connections received on the Ingress Gateway and their negotiated TLS versions. The versions can be TLSv1.3 or TLSv1.2
Type	Gauge
Dimensions	NegotiatedTLSVersion, Host, Direction, InstanceIdentifier

Table 6-25 oc_ingressgateway_backend_invocation_latency_seconds

Field	Details
Metric Name	oc_ingressgateway_backend_invocation_latency_seconds
Description	This metric captures the amount of time taken by the backend in providing response to the gateway. It starts when the request leaves the last custom filter of the application and lasts till the response reaches the custom filter of the application.
Dimensions	- quantile - InstanceIdentifier - Route_path - Method

Table 6-26 oc_ingressgateway_response_processing_latency_seconds

Field	Details
Metric Name	oc_ingressgateway_response_processing_latency_seconds
Description	This metric captures the amount of time taken for processing of the response only within Ingress gateway. It starts when the response reaches the custom filter of the application and lasts till the response is sent back to the customer.
Dimensions	- quantile - InstanceIdentifier - Route_path - Method

Table 6-27 oc_ingressgateway_request_processing_latency_seconds

Field	Details
Metric Name	oc_ingressgateway_request_processing_latency_seconds
Description	This metric tracks time between request received at first filter of SCG to request sent out from last filter of SCG.
Dimensions	- quantile - InstanceIdentifier - Route_path - Method

Table 6-28 oc_ingressgateway_congestion_system_state

Field	Details
Metric Name	oc_ingressgateway_congestion_system_state
Type	Gauge
Description	This metric provides the information of congestion level per pod.
Dimensions	NA

Table 6-29 oc_ingressgateway_congestion_cpu_state

Field	Details
Metric Name	oc_ingressgateway_congestion_cpu_state
Type	Gauge
Description	Records the CPU levels of the congestion state.
Dimensions	NA

Table 6-30 oc_ingressgateway_http_request_ratelimit_count

Field	Details
Metric Name	oc_ingressgateway_http_request_ratelimit_count
Type	Counter
Description	This metric counts the dropped messages based on rate limiting configuration. If there is no route configuration for the dropped message, the id attribute contains "NA". The "id" is set to "nomatch" for routes that do not match any routes.
Dimensions	- Id - allowed (values are true for allowed traffic and false for dropped traffic) - errorProfile

Table 6-31 oc_ingressgateway_http_request_ratelimit_reject_chain_duration_histogram

Field	Details
Metric Name	oc_ingressgateway_http_request_ratelimit_reject_chain_duration_histogram
Type	Histogram
Description	Records the reject chain duration.
Dimensions	NA

Table 6-32 oc_ingressgateway_http_request_ratelimit_reject_chain_length_histogram

Field	Details
Metric Name	oc_ingressgateway_http_request_ratelimit_reject_chain_length_histogram
Type	Histogram
Description	This metric records the number of consecutive requests rejected before a successful acceptance and adds it to the corresponding bucket. This metric provides the number of failed requests due to bursty traffic.
Dimensions	NA

Table 6-33 oc_ingressgateway_http_request_ratelimit_denied_count_total

Field	Details
Metric Name	oc_ingressgateway_http_request_ratelimit_denied_count
Type	Counter
Description	Records the details of the denied request.
Dimensions	• action • priority • congestionlevel

Table 6-34 oc_ingressgateway_system_state_duration_percentage

Field	Details
Metric Name	oc_ingressgateway_system_state_duration_percentage
Type	Gauge
Description	Records the percentage of time the system was in a particular congestion level.
Dimensions	NA

Table 6-35 oc_ingressgateway_congestion_level_total

Field	Details
Metric Name	oc_ingressgateway_congestion_level_total
Type	Counter
Description	Counts the number of times a congestion level was identified. This congestion level value varies between 0-10.
Dimensions	NA

Table 6-36 oc_ingressgateway_congestion_level_bucket_total

Field	Details
Metric Name	oc_ingressgateway_congestion_level_bucket_total
Type	Histogram
Description	Identifies the number of times the congestion level was observed.
Dimensions	NA

Table 6-37 oc_ingressgateway_congestion_cpu_percentage_bucket

Field	Details
Metric Name	oc_ingressgateway_congestion_cpu_percentage_bucket
Type	Histogram
Description	Records the CPU percentage.
Dimensions	NA

Table 6-38 oc_oauth_validation_successful_total

Field	Details
Metric Name	oc_oauth_validation_successful_total
Type	Counter
Description	This metric is pegged in the OAuth validator in the Ingress Gateway if the received OAuth token is validated successfully.
Dimensions	• issuer • subject • scope

Table 6-39 oc_oauth_validation_failure_total

Field	Details
Metric Name	oc_oauth_validation_failure_total
Type	Counter
Description	This metric will be pegged in the OAuth validator in Ingress Gateway if the validation of the received OAuth token is failed.
Dimensions	• issuer • subject • scope • reason • Status • ErrorOriginator

Table 6-40 oc_oauth_certificate_expiryStatus

Field	Details
Metric Name	oc_oauth_certificate_expiryStatus
Type	Gauge
Description	This metric is used to peg expiry date of the certificate and for raising alarms if the certificate expires within 7 or 30 days.
Dimensions	• id • certificateName • secretName

Table 6-41 oc_oauth_certificate_loadStatus

Field	Details
Metric Name	oc_oauth_certificate_loadStatus
Type	Gauge
Description	This metric is used to peg whether the given certificate can be loaded from secret or not. If the certificate can be loaded then 0 is pegged otherwise 1 is pegged. This metric is also used for raising alarms when certificate cannot be loaded.
Dimensions	• id • certificateName • secretName

Table 6-42 oc_oauth_request_failed_cert_expiry

Field	Details
Metric Name	oc_oauth_request_failed_cert_expiry
Type	Counter
Description	This metric is used to keep track of the number of requests with keyId in the token that failed due to certificate expiry. It is pegged when OAuth validator module throws an OAuth custom exception due to certificate expiry for an incoming request.
Dimensions	• target nf type • target nf instance id • consumer nf instance id • nrf instance id • service name of nf producer service • key id

Table 6-43 oc_oauth_keyid_count

Field	Details
Metric Name	oc_oauth_keyid_count
Type	Counter
Description	This metric is used to keep track of the number of requests received with keyId in the token. This metric is pegged when a request with an access token containing keyId in the header comes to OAuth validator and the validation is successful.
Dimensions	• target nf type • target nf instance id • consumer nf instance id • nrf instance id • service name of nf producer service • key id

6.2.2 Egress API Gateway Metrics

The following metrics are applicable to Egress API Gateway.

Table 6-44 Total Egress Requests

Field	Details
Metric Name	oc_ingressgateway_http_requests_total{Method="POST/PUT/GET/DELETE/PATCH", Scheme="HTTP/HTTPS",container="ingressgateway-sig"}
Description	oc_egressgateway_http_requests_total Metric to capture the request count reaches the Egress gateway from the application microservice and pegs with Direction as egress. Also, when the request goes out of egress gateway and pegs with Direction as egressOut.
Type	Counter
Dimensions	Method NFType NFServiceType Host HttpVersion Scheme

Table 6-45 Total Egress Responses

Field	Details
Metric Name	oc_egressgateway_http_responses_total{Method="POST/PUT/GET/DELETE",Status=~"2.* / 4.*"}
Description	Metric to capture when Egress Gateway sends response back to backend NF microservice and pegs with Direction as egress. Also, when the response is received Egress Gateway and pegs with Direction as egressOut.
Type	Counter
Dimensions	Status Method NFType NFServiceType Host HttpVersion Scheme

Table 6-46 Connection Failures Egress Gateway

Field	Details
Metric Name	oc_egressgateway_connection_failure_total
Description	Metric to capture failure when the destination is not reachable by Egress Gateway. Here, the destination is producer NF.
Type	Counter
Dimensions	Host Port InstanceIdentifier Direction error_reason

Table 6-47 Egress Gateway Request Processing Latency

Field	Details
Metric Name	oc_egressgateway_request_processing_latency_seconds
Description	Metric to capture the amount of time taken for processing of the request only within Egress gateway
Type	Timer
Dimensions	quantile InstanceIdentifier

Table 6-48 Egress Gateway Request Processing Latency

Field	Details
Metric Name	oc_egressgateway_routing_invalid_config_detected
Description	Metric to indicate whether the filter configuration done for a specific route is valid or not. This covers all filters including SBI filter and increments if any misconfiguration is found related to any configured filter.
Type	Timer
Dimensions	Not Applicable

Table 6-49 oc_egressgateway_outgoing_tls_connections

Field	Details
Description	Number of TLS connections received on the Egress Gateway and their negotiated TLS versions. The versions can be TLSv1.3 or TLSv1.2
Type	Gauge
Dimensions	NegotiatedTLSVersion, Host, Direction, InstanceIdentifier

6.3 nudr-dr-service Metrics

The following metrics are applicable to the nudr-dr service of UDR.

Table 6-50 SLF default GroupId

Field	Details
Metric Name	slf_sucess_txn_default_grp_id
Description	The number of requests for which the SLF responds with default GroupId for Lookup
Type	Counter
Dimensions	Not Applicable

Table 6-51 EIR Default Response

Field	Details
Metric Name	eir_txn_default_response_sent
Description	The number of EIR EIC responses using default response
Type	Counter

Table 6-51 (Cont.) EIR Default Response

Field	Details
Dimensions	Interface

Table 6-52 UDR Total Requests

Field	Details
Metric Name	udr_rest_all_requests_total
Description	Total number of requests received by nudr-drservice (All UDR operations).
Type	Counter
Dimensions	sbi_correlation_header(yes/no) key(imsi/msisdn/nai/extid/accountid)

Table 6-53 UDR Total Responses

Field	Details
Metric Name	udr_rest_all_response_total
Description	Total number of responses sent by nudr-drservice (All UDR operations).
Type	Counter
Dimensions	sbi_correlation_header(yes/no) key(imsi/msisdn/nai/extid/accountid) status(failure/success)

Table 6-54 UDR Total Success Responses

Field	Details
Metric Name	udr_rest_success_response_total
Description	Total number of success responses sent by nudr-drservice (All UDR operations).
Type	Counter
Dimensions	Not Applicable

Table 6-55 UDR Total Failure Responses

Field	Details
Metric Name	udr_rest_failure_response_total
Description	Total number of failure responses sent by nudr-drservice (All UDR operations).
Type	Counter
Dimensions	Not Applicable

Table 6-56 UDR Put Requests

Field	Details
Metric Name	udr_rest_request_total{Method="PUT/PATCH/GET/DELETE/POST"}
Description	Total number of PUT requests received by nudr-drservice (UDR related).
Type	Counter

Table 6-56 (Cont.) UDR Put Requests

Field	Details
Dimensions	Method sbi_correlation_header key status

Table 6-57 UDR Put Success Responses

Field	Details
Metric Name	udr_rest_response_success_total{Method="PUT"}
Description	Total number of PUT success responses sent by nudr-dr-service (UDR related).
Type	Counter
Dimensions	Method

Table 6-58 UDR Patch Success Responses

Field	Details
Metric Name	udr_rest_response_success_total{Method="PATCH/GET/DELETE/POST"}
Description	Total number of PATCH success responses sent by nudr-dr-service (UDR related).
Type	Counter
Dimensions	Method

Table 6-59 UDR Patch Failure Responses

Field	Details
Metric Name	udr_rest_response_failure_total{Method="PATCH/GET/DELETE/POST"}
Description	Total number of PATCH failure responses sent by nudr-dr-service (UDR related).
Type	Counter
Dimensions	Method

Table 6-60 UDR Schema Validation Success

Field	Details
Metric Name	udr_schema_validation_success_total
Description	Total success count of schema validations done for all operations with body.
Type	Counter
Dimensions	Not Applicable

Table 6-61 UDR Schema Validation Failure

Field	Details
Metric Name	udr_schema_validation_failure_total

Table 6-61 (Cont.) UDR Schema Validation Failure

Field	Details
Description	Total failure count of schema validations done for all operations with body.
Type	Counter
Dimensions	Not Applicable

Table 6-62 Db Operations Total

Field	Details
Metric Name	udr_db_operations_total{Method="READ/UPDATE"}
Description	Total number of database reads and writes performed by nldr-dr-provservice on the back end NDB Cluster.
Type	Counter
Dimensions	Method

Table 6-63 Db Operations Success

Field	Details
Metric Name	udr_db_operations_success_total{Method="READ/UPDATE"}
Description	Total number of successful DB reads performed by nldr-dr-service on the back end NDB Cluster.
Type	Counter
Dimensions	Method

Table 6-64 Db Operations Failure

Field	Details
Metric Name	udr_db_operations_failure_total{Method="READ/UPDATE"}
Description	Total number of failed DB reads performed by nldr-dr-service on the back end NDB Cluster.
Type	Counter
Dimensions	Method

Table 6-65 Db Total Txn Attempted

Field	Details
Metric Name	udr_total_transactions_total
Description	Total number of successful DB transactions (includes all DB operations) performed by nldr-dr-service on the back end NDB Cluster.
Type	Counter
Dimensions	Not Applicable

Table 6-66 DB Txn Success

Field	Details
Metric Name	udr_successful_transactions_total
Description	Total number of failed DB transactions (includes all DB operations) performed by nudr-drservice on the back end NDB Cluster.
Type	Counter
Dimensions	Not Applicable

Table 6-67 UDSF Total Requests

Field	Details
Metric Name	udsf_rest_all_requests_total
Description	Total number of requests received by nudr-drservice (All UDSF operations).
Type	Counter
Dimensions	Not Applicable

Table 6-68 UDSF Total Response

Field	Details
Metric Name	udsf_rest_allresponse_total
Description	Total number of responses sent by nudr-drservice (All UDSF operations).
Type	Counter
Dimensions	Not Applicable

Table 6-69 UDSF Total Success Responses

Field	Details
Metric Name	udsf_rest_success_response_total
Description	Total number of success responses sent by nudr-drservice (All UDSF operations).
Type	Counter
Dimensions	Not Applicable

Table 6-70 UDSF Total Failure Responses

Field	Details
Metric Name	UDSF Total Failure Responses
Description	Total number of failure responses sent by nudr-drservice (All UDSF operations).
Type	Counter
Dimensions	Not Applicable

Table 6-71 UDSF Put Requests

Field	Details
Metric Name	udsf_rest_request_total{Method="PUT/GET/DELETE"}

Table 6-71 (Cont.) UDSF Put Requests

Field	Details
Description	Total number of PUT requests received by nudr-drservice (UDSF related).
Type	Counter
Dimensions	Method

Table 6-72 UDSF Put Success Responses

Field	Details
Metric Name	udsf_rest_response_success_total{Method="PUT/GET/DELETE"}
Description	Total number of PUT success responses sent by nudr-drservice (UDSF related).
Type	Counter
Dimensions	Method

Table 6-73 UDSF Put Failure Responses

Field	Details
Metric Name	udsf_rest_response_failure_total{Method="PUT/GET/DELETE"}
Description	Total number of PUT failure responses sent by nudr-drservice (UDSF related).
Type	Counter
Dimensions	Method

Table 6-74 UDR Subscription Requests

Field	Details
Metric Name	nudr_dr_all_subscription_requests_total
Description	Total number of subscription requests received by nudr-drservice (All operations).
Type	Counter
Dimensions	Not Applicable

Table 6-75 UDR Subscription Request Successful

Field	Details
Metric Name	nudr_dr_all_subscription_responses_total
Description	Total number of successful subscription request processed successfully by nudr-drservice (All operations).
Type	Counter
Dimensions	Not Applicable

Table 6-76 UDR Subscription Response Successful

Field	Details
Metric Name	nudr_dr_all_subscription_successful_responses_total
Description	Total number of successful subscription response sent by nudr-drservice (All operations).

Table 6-76 (Cont.) UDR Subscription Response Successful

Field	Details
Type	Counter
Dimensions	Not Applicable

Table 6-77 UDR Subscription Response Failure

Field	Details
Metric Name	nudr_dr_all_subscription_failed_responses_total
Description	Total number of successful subscription response sent by nudr-dr-service (All operations).
Type	Counter
Dimensions	Not Applicable

Table 6-78 UDR Post Subscription Requests

Field	Details
Metric Name	nudr_dr_subscription_request_total{Method="POST/PUT/GET/DELETE"}
Description	Total number of POST subscription requests received by nudr-dr-service.
Type	Counter
Dimensions	Method

Table 6-79 UDR Post Subscription Response Success

Field	Details
Metric Name	nudr_dr_subscription_successful_response_total{Method="POST/PUT/GET/DELETE"}
Description	Total number of successful POST subscription response sent by nudr-dr-service
Type	Counter
Dimensions	Method

Table 6-80 UDR Post Subscription Response Failure

Field	Details
Metric Name	nudr_dr_subscription_failed_response_total{Method="POST/PUT/GET/DELETE"}
Description	Total number of failed POST Subscription response sent by nudr-dr-service
Type	Counter
Dimensions	Method

Table 6-81 UDR Notification Requests

Field	Details
Metric Name	nudr_dr_notif_signal_sent_total
Description	Total notification sent successfully by nudr-dr-service to nudr-notify-service.
Type	Counter

Table 6-81 (Cont.) UDR Notification Requests

Field	Details
Dimensions	Not Applicable

Table 6-82 UDR Notification Failure

Field	Details
Metric Name	nudr_dr_notif_signal_failed_notifications_total
Description	Total notification failed to send to nudr-notify-service.
Type	Counter
Dimensions	Not Applicable

Table 6-83 UDR Update Notification Success

Field	Details
Metric Name	nudr_dr_notif_signal_notification_update_success_total
Description	Total update notifications sent successfully to nudr-notify-service.
Type	Counter
Dimensions	Method

Table 6-84 UDR Update Notification Failure

Field	Details
Metric Name	nudr_dr_notif_signal_notification_update_failed_total
Description	Total update notifications failed to send to nudr-notify-service.
Type	Counter
Dimensions	Method

Table 6-85 UDR Delete Notification Success

Field	Details
Metric Name	nudr_dr_notif_signal_notification_delete_success_total
Description	Total delete notifications sent successfully to nudr-notify-service
Type	Counter
Dimensions	Method

Table 6-86 UDR Delete Notification Failure

Field	Details
Metric Name	nudr_dr_notif_signal_notification_delete_failed_total
Description	Total delete notifications failed to send to nudr-notify-service
Type	Counter
Dimensions	Method

Table 6-87 UDR NF Type Specific Metrics - Method wise

Field	Details
Metric Name	udr_nftype_specific_requests{NFType="PCF/UDM/SLF/NEF/EIR",Method="GET/POST/PUT/DELETE/PATCH",ProvRequest="N",subtype="SLFGroupName/NFGroupIdMap/ExposureAMData/ExposurePSMData/subs-to-notify/EquipmentStatus"}
Description	Total number of requests that UDR receives from different NF types along with method types.
Type	Counter
Dimensions	Method NFType ProvRequest subtype

Table 6-88 UDR NF Type Specific Metrics - Method wise

Field	Details
Metric Name	udr_nftype_specific_requests_status{NFType="PCF/UDM/SLF/NEF/EIR",Method="GET/POST/PUT/DELETE/PATCH",ProvRequest="N",subtype="SLFGroupName/NFGroupIdMap/ExposureAMData/ExposurePSMData/subs-to-notify/EquipmentStatus"}
Description	Total number of success responses that UDR sends to different NFs along with method types
Type	Counter
Dimensions	Method NFType ProvRequest subtype

Table 6-89 UDR NF Type Subscription Metrics - Method Wise

Field	Details
Metric Name	nudr_dr_nf_type_subscription_request{NFType="PCF/UDM/NEF",Method="POST/PUT/DELETE/GET"}
Description	Total number of subscription requests for each NF type as well as each method type.
Type	Counter
Dimensions	Method NFType

Table 6-90 UDR NF Type Subscription Metrics - Success Response

Field	Details
Metric Name	nudr_dr_nf_type_subscription_request_status{NFType="PCF/UDM/NEF",StatusCode=~"2.*"/"4.*",Method="POST/PUT/DELETE/GET/PATCH"}
Description	Total number of success responses of subscriptions.
Type	Counter

Table 6-90 (Cont.) UDR NF Type Subscription Metrics - Success Response

Field	Details
Dimensions	Method NFType StatusCode

Table 6-91 Config Service Update

Field	Details
Metric Name	Total number of config updates received by nudr-config microservices
Description	If the configuration change is successfully (2xx) received or throws an error (4xx).
Type	Counter
Dimensions	StatusCode

Table 6-92 VSA Specific Requests

Field	Details
Metric Name	udr_vsa_specific_requests_total{Method="POST/PUT/DELETE/GET/PATCH"}
Description	Total number of VSA requests method wise
Type	Counter
Dimensions	Method NFType ResourceType

Table 6-93 udr_request_processing_time

Field	Details
Metric Name	udr_request_processing_time
Description	Average time taken by nudr-dr service to process the request including cnDBtier and nudr-dr service.
Type	Timer
Dimensions	Not Applicable

Table 6-94 udr_db_processing_time

Field	Details
Metric Name	udr_db_processing_time
Description	Average time taken to process the database operations excluding the application logic.
Type	Timer
Dimensions	Not Applicable

Table 6-95 slf_sucess_txn_default_grp_id

Field	Details
Metric Name	slf_sucess_txn_default_grp_id
Description	The number of requests for which the SLF reponds with default GroupId for Lookup.
Type	Counter
Dimensions	Not Applicable

Table 6-96 nudr_dr_subscription_expired

Field	Details
Metric Name	nudr_dr_subscription_expired
Description	This metric will be logged when the subscription is expired.
Type	Counter
Dimensions	NFType

Table 6-97 udr_subscriber_not_found

Field	Details
Metric Name	udr_subscriber_not_found{resource_name=~"nudr-dr nudr-group-id-map nudsf-dr"}
Description	This metric will be logged whenever subscriber is not present.
Type	Counter
Dimensions	resource-name

Table 6-98 eir_txn_default

Field	Details
Metric Name	eir_txn_default_response_sent
Description	The number of EIR EIC responses using default response.
Type	Counter
Dimensions	Not Applicable

Table 6-99 Default SM Policy SNSSAI Data Creation

Field	Details
Metric Name	udr_default_smPolicySnssaiData_creation_total{Method="PUT/PATCH"}
Description	Total number of requests for which default value of smPolicySnssaiData is applied.
Type	Counter
Dimensions	Not Applicable

Table 6-100 Auto Enrollment of Subscriber

Field	Details
Metric Name	total_auto_enrol_requests_total

Table 6-100 (Cont.) Auto Enrollment of Subscriber

Field	Details
Description	The total number of auto enrollment requests.
Type	Counter
Dimensions	Not Applicable

Table 6-101 ETag Rest Request

Field	Details
Metric Name	udr_etag_rest_request_total{Method="PUT/GET/PATCH/DELETE",etag_header="if-match/if-none-match"}
Description	Total number of Rest request received for which ETag is enabled.
Type	Counter
Dimensions	Not Applicable

Table 6-102 ETag Rest Response

Field	Details
Metric Name	udr_etag_rest_response_total{Method(PUT,GET,PATCH,DELETE),status(200,201,204,412,304)}
Description	Total number of responses sent for which ETag is enabled.
Type	Counter
Dimensions	Not Applicable

Table 6-103 EIR Default Response IMEI Not Found

Field	Details
Metric Name	eir_default_response_imei_not_found
Description	The number of EIR and EIC responses using default response when IMEI or PEI is not found in the database.
Type	Counter
Dimensions	Not Applicable

Table 6-104 Suppress Notifications

Field	Details
Metric Name	suppress_user_agent_not_found_total
Description	This metric provides the total number of requests without User-Agent header when the feature is enabled.
Type	Counter
Dimensions	NA

Table 6-105 udr_subscriptions_recreated

Field	Details
Metric Name	udr_subscriptions_recreated
Description	This metric provides the total number of POST requests received for the existing subscription when allowSubscriptionRecreation is set to true.
Type	Counter
Dimensions	NA

6.4 nudr-dr-provservice Metrics

The following metrics are applicable to the nudr-dr-provservice of UDR.

Table 6-106 UDR Total Requests

Field	Details
Metric Name	udr_rest_all_requests_total
Description	Total number of requests received by nudr-dr-provservice (All UDR operations)
Type	Counter
Dimensions	sbi_correlation_header(no)key(imsi/msisdn/nai/extid/accountid)

Table 6-107 UDR Total Responses

Field	Details
Metric Name	udr_rest_all_response_total
Description	Total number of responses sent by nudr-dr-provservice (All UDR operations)
Type	Counter
Dimensions	sbi_correlation_header(no) key(imsi/msisdn/nai/extid/accountid) status(failure/success)

Table 6-108 UDR TotalSuccess Responses

Field	Details
Metric Name	udr_rest_success_response_total
Description	Total number of success responses sent by nudr-dr-provservice (All UDR operations)
Type	Counter
Dimensions	Not Applicable

Table 6-109 UDR Total Failure Responses

Field	Details
Metric Name	udr_rest_failure_response_total
Description	Total number of failure responses sent by nudr-dr-provservice (All UDR operations)
Type	Counter
Dimensions	Not Applicable

Table 6-110 UDR Put Requests

Field	Details
Metric Name	udr_rest_request_total{Method="PUT/PATCH/GET/DELETE/POST"}
Description	Total number of PUT requests received by nudr-dr-provservice (UDR related)
Type	Counter
Dimensions	Method sbi_correlation_header key status

Table 6-111 UDR Put Success Responses

Field	Details
Metric Name	udr_rest_response_success_total{Method="PUT"}
Description	Total number of PUT success responses sent by nudr-dr-provservice (UDR related)
Type	Counter
Dimensions	Method

Table 6-112 UDR Put Failure Responses

Field	Details
Metric Name	udr_rest_response_failure_total{Method="PUT"}
Description	Total number of PUT failure responses sent by nudr-dr-provservice (UDR related)
Type	Counter
Dimensions	Method

Table 6-113 UDR Patch Success Responses

Field	Details
Metric Name	udr_rest_response_success_total{Method="PATCH/GET/DELETE/POST"}
Description	Total number of PATCH success responses sent by nudr-dr-provservice (UDR related)
Type	Counter
Dimensions	Method

Table 6-114 UDR Patch Failure Responses

Field	Details
Metric Name	udr_rest_response_failure_total{Method="PATCH/GET/DELETE/POST"}
Description	Total number of PATCH failure responses sent by nudr-dr-provservice (UDR related)
Type	Counter
Dimensions	Method

Table 6-115 UDR Schema Validation Success

Field	Details
Metric Name	udr_schema_validation_success_total
Description	Total success count of schema validations done for all operations with body.
Type	Counter
Dimensions	Not Applicable

Table 6-116 UDR Schema Validation Failure

Field	Details
Metric Name	udr_schema_validation_failure_total
Description	Total failure count of schema validations done for all operations with body.
Type	Counter
Dimensions	Not Applicable

Table 6-117 UDR Put Schema Success

Field	Details
Metric Name	udr_schema_mgm_operations_success_total
Description	Total number of successful schema updates processed by nudr-dr-provservice
Type	Counter
Dimensions	Method (PUT/GET/DELETE) RESOURCE (POLICY_DATA/ SUBSCRIPTION_DATA/ PROFILE_DATA/SLF_GROUP) EXPOSURE_DATA

Table 6-118 UDR Put Schema Failure

Field	Details
Metric Name	udr_schema_mgm_operations_failure_total
Description	Total number of failure schema updates processed by nudr-dr-provservice
Type	Counter
Dimensions	Method (PUT/GET/DELETE) RESOURCE (POLICY_DATA/ SUBSCRIPTION_DATA/ PROFILE_DATA/SLF_GROUP) EXPOSURE_DATA

Table 6-119 Db Operations Total

Field	Details
Metric Name	udr_db_operations_total{Method="READ/UPDATE"}

Table 6-119 (Cont.) Db Operations Total

Field	Details
Description	Total number of database reads and writes performed by nudr-dr-provservice on the back end NDB Cluster.
Type	Counter
Dimensions	Method

Table 6-120 Db Operations Success

Field	Details
Metric Name	udr_db_operations_success_total{Method="READ/UPDATE"}
Description	Total number of successful DB reads performed by nudr-dr-service on the back end NDB Cluster.
Type	Counter
Dimensions	Method

Table 6-121 Db Operations Failure

Field	Details
Metric Name	udr_db_operations_failure_total{Method="READ/UPDATE"}
Description	Total number of failed DB reads performed by nudr-dr-service on the back end NDB Cluster.
Type	Counter
Dimensions	Method

Table 6-122 Db Total Txn Attempted

Field	Details
Metric Name	udr_total_transactions_total
Description	Total number of successful DB transactions (includes all DB operations) performed by nudr-dr-service on the back end NDB Cluster.
Type	Counter
Dimensions	Not Applicable

Table 6-123 DB Txn Success

Field	Details
Metric Name	udr_successful_transactions_total
Description	Total number of failed DB transactions (includes all DB operations) performed by nudr-dr-service on the back end NDB Cluster.
Type	Counter
Dimensions	Not Applicable

Table 6-124 UDR Subscription Requests

Field	Details
Metric Name	nudr_dr_all_subscription_requests_total
Description	Total number of subscription requests received by nudr-dr-provservice (All operations)
Type	Counter
Dimensions	Not Applicable

Table 6-125 UDR Subscription Request Successful

Field	Details
Metric Name	nudr_dr_all_subscription_responses_total
Description	Total number of successful subscription request processed successfully by nudr-dr-provservice (All operations)
Type	Counter
Dimensions	Not Applicable

Table 6-126 UDR Subscription Response Failure

Field	Details
Metric Name	nudr_dr_all_subscription_failed_responses_total
Description	Total number of failed subscription response sent by nudr-dr-provservice (All operations)
Type	Counter
Dimensions	Not Applicable

Table 6-127 UDR Post Subscription Requests

Field	Details
Metric Name	nudr_dr_subscription_request_total{Method="POST/PUT/GET/DELETE"}
Description	Total number of successful POST Subscription requests processed by nudr-dr-provservice
Type	Counter
Dimensions	Method

Table 6-128 UDR Post Subscription Response Success

Field	Details
Metric Name	nudr_dr_subscription_successful_response_total{Method="POST/PUT/GET/DELETE"}
Description	Total number of successful POST Subscription response sent by nudr-dr-provservice
Type	Counter
Dimensions	Method

Table 6-129 UDR Post Subscription Response Failure

Field	Details
Metric Name	nudr_dr_subscription_failed_response_total{Method="POST/PUT/GET/DELETE"}

Table 6-129 (Cont.) UDR Post Subscription Response Failure

Field	Details
Description	Total number of failed POST Subscription response sent by nudr-dr-provservice
Type	Counter
Dimensions	Method

Table 6-130 UDR Notification Requests

Field	Details
Metric Name	nudr_dr_notif_signal_sent_total
Description	Total number of notifications sent successfully by nudr-dr-provservice to nudr-notify-service
Type	Counter
Dimensions	Not Applicable

Table 6-131 UDR Notification Failure

Field	Details
Metric Name	nudr_dr_notif_signal_failed_notifications_total
Description	Total number of notifications failed to send to nudr-notify-service
Type	Counter
Dimensions	Not Applicable

Table 6-132 UDR Update Notification Success

Field	Details
Metric Name	nudr_dr_notif_signal_notification_update_success_total
Description	Total number of update notifications sent successfully to nudr-notify-service
Type	Counter
Dimensions	Method

Table 6-133 UDR Update Notification Failure

Field	Details
Metric Name	nudr_dr_notif_signal_notification_update_failed_total
Description	Total number of update notifications failed to send to nudr-notify-service
Type	Counter
Dimensions	Method

Table 6-134 UDR Delete Notification Success

Field	Details
Metric Name	nudr_dr_notif_signal_notification_delete_success_total
Description	Total number of delete notifications sent successfully to nudr-notify-service

Table 6-134 (Cont.) UDR Delete Notification Success

Field	Details
Type	Counter
Dimensions	Method

Table 6-135 UDR Delete Notification Failure

Field	Details
Metric Name	nudr_dr_notif_signal_notification_delete_failed_total
Description	Total number of delete notifications failed to send to nudr-notify-service
Type	Counter
Dimensions	Method

Table 6-136 UDR NF Type Specific Metrics - Method wise

Field	Details
Metric Name	udr_nftype_specific_requests{NFType="PCF/UDM/SLF/NEF/EIR",Method="GET/POST/PUT/DELETE/PATCH", ProvRequest="Y", subtype="SLFGroupName/NFGroupIdMap/EquipmentStatus/EIRProfileUpdate"}
Description	Total number of requests that UDR receives from different NF types along with method types
Type	Counter
Dimensions	Method NFType ProvRequest subtype

Table 6-137 UDR NF Type Specific Metrics - Success Response, Subtype wise, Method wise, and Provisioning Request

Field	Details
Metric Name	udr_nftype_specific_requests_status{NFType="PCF/UDM/SLF/NEF/EIR",StatusCode=~"2.*/*4.*/*5.*", ProvRequest="Y", Method="GET/POST/PUT/PATCH/DELETE", subtype="SLFGroupName/NFGroupIdMap/EquipmentStatus/EIRProfileUpdate"}
Description	Total number of success responses for SLF based on subtype, method wise
Type	Counter
Dimensions	Method NFType ProvRequest subtype StatusCode

Table 6-138 UDR NF Type Subscription Metrics - Method Wise

Field	Details
Metric Name	nudr_dr_nf_type_subscription_request{NFType="PCF/UDM/NEF",Method="POST/PUT/DELETE/GET"}
Description	Total number of subscription requests for each NF type as well as each method type
Type	Counter
Dimensions	Method NFType

Table 6-139 UDR NF Type Subscription Metrics - Success Response

Field	Details
Metric Name	nudr_dr_nf_type_subscription_request_status{NFType="PCF/UDM/NEF",StatusCode=~"2.*"/4.*", Method="POST/PUT/DELETE/GET/PATCH"}
Description	Total number of success responses of subscriptions
Type	Counter
Dimensions	Method NFType StatusCode

Table 6-140 Config Service Update

Field	Details
Metric Name	nudr_dr_config_service_update{StatusCode:200/400}
Description	If the configuration change is successfully(2xx) received or throws a error(4xx)
Type	Counter
Dimensions	StatusCode

Table 6-141 VSA Specific Requests Total

Field	Details
Metric Name	udr_vsa_specific_requests_total{Method="POST/PUT/DELETE/GET/PATCH"}
Description	Total number of VSA requests on the basis of method
Type	Counter
Dimensions	Method NFType ResourceType

Table 6-142 VSA Specific Requests Status

Field	Details
Metric Name	udr_vsa_specific_requests_status_total{StatusCode=~"2.*"/4.*",Method="POST/PUT/DELETE/GET/PATCH"}
Description	Total number of VSA requests Success(2xx)/Failure(4xx) Response method wise
Type	Counter

Table 6-142 (Cont.) VSA Specific Requests Status

Field	Details
Dimensions	Method NFType ResourceType StatusCode

Table 6-143 Request Processing Time

Field	Details
Metric Name	udr_request_processing_time
Description	Time to process UDR request. This is measured from when the request is received to when the response is sent
Type	Timer
Dimensions	Not Applicable

Table 6-144 Database Processing Time

Field	Details
Metric Name	udr_db_processing_time
Description	Time to process a DB query. This is measured from when the database method starts to the db method ends.
Type	Timer
Dimensions	Not Applicable

Table 6-145 Subscription Expired

Field	Details
Metric Name	nudr_dr_subscription_expired
Description	This metric is logged when the subscription is expired.
Type	Counter
Dimensions	NFType

Table 6-146 Subscriber Not Found

Field	Details
Metric Name	udr_subscriber_not_found{resource_name=~"nudr-dr-prov nudr-group-id-map-prov slf-group-prov"}
Description	This metric is logged whenever the subscriber is not present.
Type	Counter
Dimensions	resource-name

6.5 nudr-notify-service Metrics

The following metrics are applicable to the nudr-notify-service of UDR.

Table 6-147 UDR Notification Requests

Field	Details
Metric Name	nudr_notif_notifications_signals_received_total
Description	Total number of notifications received on nudr-notify-service from nudr-drservice.
Type	Counter
Dimensions	Not Applicable

Table 6-148 UDR Failed Notifications Processing

Field	Details
Metric Name	nudr_notif_notifications_processing_failed_total
Description	Total number of notification failure processing on nudr-notify-service.
Type	Counter
Dimensions	Not Applicable

Table 6-149 UDR Notifications Sent

Field	Details
Metric Name	nudr_notif_notifications_sent_total
Description	Total number of notifications sent to network.
Type	Counter
Dimensions	sbi_correlation_header(yes/no) key(imsi/msisdn/nai/extid/accountid)

Table 6-150 UDR Send Notification Failures

Field	Details
Metric Name	nudr_notif_notifications_send_fail_total
Description	Total number of notifications failed to send to network.
Type	Counter
Dimensions	Not Applicable

Table 6-151 UDR Notifications Marked Retry

Field	Details
Metric Name	nudr_notif_notifications_marked_for_retry_total
Description	Total number of notifications marked for retry after notification failure response.
Type	Counter
Dimensions	Not Applicable

Table 6-152 UDR Notification Responses

Field	Details
Metric Name	nudr_notif_notifications_ack_total
Description	Total number of responses for notifications sent to network.

Table 6-152 (Cont.) UDR Notification Responses

Field	Details
Type	Counter
Dimensions	Not Applicable

Table 6-153 UDR Notification Success Responses

Field	Details
Metric Name	nudr_notif_notifications_ack_2xx_total
Description	Total number of success responses for notifications sent to network.
Type	Counter
Dimensions	Not Applicable

Table 6-154 UDR Notification 4xx Responses

Field	Details
Metric Name	nudr_notif_notifications_ack_4xx_total
Description	Total number of 4xx error responses for notifications sent to network.
Type	Counter
Dimensions	Not Applicable

Table 6-155 UDR Notification 5xx Responses

Field	Details
Metric Name	nudr_notif_notifications_ack_5xx_total
Description	Total number of 5xx error responses for notifications sent to network.
Type	Counter
Dimensions	Not Applicable

Table 6-156 UDR NF Type Notification Metric

Field	Details
Metric Name	nudr_notif_notifications_nftype_signals_received{NFType="UDM/PCF",Method="UPDATE/DELETE"}
Description	Total number of notifications received to sent out to UDM or PCF with update or delete requests.
Type	Counter
Dimensions	Method NFType

Table 6-157 Config Service Update

Field	Details
Metric Name	nudr_notify_config_service_update{StatusCode:200/400}

Table 6-157 (Cont.) Config Service Update

Field	Details
Description	If the configuration change is received successfully(200) or throws error while processing (400)
Type	Counter
Dimensions	StatusCode

Table 6-158 VSA Notification Received

Field	Details
Metric Name	nudr_notif_notifications_vsa_signals_received {UPDATE, DELETE}
Description	Total number of notifications update received by Nudr-notify when the sm-data/vsa is updated.
Type	Counter
Dimensions	Method NFType ResourceType

Table 6-159 Subscriptions Expired Count

Field	Details
Metric Name	nudr_notif_subscription_expired
Description	Total number of expired notifications for the subscription requests.
Type	Counter
Dimensions	NFType Notification Type

Table 6-160 Notification Count with ETag Header

Field	Details
Metric Name	nudr_notif_etag_notifications_sent_total
Description	Total number of notifications for which ETAG is sent from Nudr-dr/Nudr-dr-prov microservices.
Type	Counter
Dimensions	etag (yes/no)

Table 6-161 Notify Supress Notifications

Field	Details
Metric Name	nudr_notif_suppress_notifications_total
Description	This metrics indicates number of suppressed notifications.
Type	Counter
Dimensions	NA

6.6 nudr-diameterproxy-service Metrics

The following metrics are applicable to the nudr-diameterproxy-service of UDR.

Table 6-162 Diameter proxy Request Messages

Field	Details
Metric Name	nudr_diameterproxy_total_requests_total
Description	The total number of requests received on the diameterproxy microservice.
Type	Counter
Dimensions	Not Applicable

Table 6-163 Diameter proxy Response Messages

Field	Details
Metric Name	nudr_diameterproxy_total_response_total
Description	Total number of Sh message responses from diameter peer
Type	Counter
Dimensions	ResultCode

Table 6-164 Diameter proxy Success Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{ResultCode="2001"}
Description	Total number of successful Sh message responses from diameter peer.
Type	Counter
Dimensions	MessageType ResultCode

Table 6-165 Diameter proxy Failure Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{ResultCode!="2001"}
Description	Total number of failure Sh message responses from diameter peer
Type	Counter
Dimensions	MessageType ResultCode

Table 6-166 Diameter proxy Decode Failed Messages

Field	Details
Metric Name	nudr_diameterproxy_message_decoding_failed_total
Description	The total number of messages failed while decoding and invalid AVPs in the message.
Type	Counter
Dimensions	Not Applicable

Table 6-167 Diameter proxy Discarded Messages

Field	Details
Metric Name	nudr_diameterproxy_request_discarded_total
Description	The total number of discarded messages due to invalid command codes.
Type	Counter
Dimensions	Not Applicable

Table 6-168 Diameter proxy UDA Success Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="UDA", ResultCode="2001"}
Description	The total number of success User Data Answer messages received on diameterproxy.
Type	Counter
Dimensions	MessageType ResultCode

Table 6-169 Diameter proxy UDA Failure Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="UDA", ResultCode != "2001"}
Description	The total number of failure User Data Answer messages received on diameterproxy.
Type	Counter
Dimensions	MessageType ResultCode

Table 6-170 Diameterproxy_UDR_Rejected_By_Unknown_User

Field	Details
Metric Name	nudr_diameterproxy_request_rejected_unknown_user_total
Description	The total number User data requests messages failed because of Unknown user.
Type	Counter
Dimensions	Not Applicable

Table 6-171 Diameterproxy_UDR_Rejected_By_Invalid_Service_Indication

Field	Details
Metric Name	nudr_diameterproxy_rejected_invalid_service_indication_requests_total{MessageType="UD R"}
Description	The total number User Data request messages failed because of invalid service passed.
Type	Counter
Dimensions	MessageType

Table 6-172 Diameterproxy_PUR_Messages

Field	Details
Metric Name	nudr_diameterproxy_requests_total{MessageType="PUR"}
Description	The total number of profile update request messages received on diameterproxy.
Type	Counter
Dimensions	MessageType

Table 6-173 Diameterproxy_PUA_Success_Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="PUA", ResultCode="2001"}
Description	The total number of success profile data update requests
Type	Counter
Dimensions	MessageType ResultCode

Table 6-174 Diameterproxy_PUA_Failure_Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="PUA", ResultCode != "2001"}
Description	The total number of failure profile data update requests
Type	Counter
Dimensions	MessageType ResultCode

Table 6-175 Diameterproxy_PUR_Rejected_By_Unknown_Use

Field	Details
Metric Name	nudr_diameterproxy_request_rejected_unknown_user_total
Description	The total number of PUR messages rejected by unknown user(Subscriber doesn't exists).
Type	Counter
Dimensions	Not Applicable

Table 6-176 Diameterproxy_PUR_Rejected_By_Invalid_Service_Indication

Field	Details
Metric Name	nudr_diameterproxy_rejected_invalid_service_indication_requests_total{MessageType="PUR"}
Description	The total number of failed profile update requests with passing invalid service indication.
Type	Counter
Dimensions	MessageType

Table 6-177 Diameterproxy_SNR_Messages

Field	Details
Metric Name	nudr_diameterproxy_requests_total{MessageType="SNR"}
Description	The total number of subscriber notification request messages received on diameterproxy.
Type	Counter
Dimensions	MessageType

Table 6-178 Diameterproxy_SNA_Success_Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="SNA", ResultCode="2001"}
Description	The total number of Success subscriber notification messages.
Type	Counter
Dimensions	MessageType ResultCode

Table 6-179 Diameterproxy_SNA_Failure_Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="SNA", ResultCode != "2001"}
Description	The total number of failure subscriber notification messages
Type	Counter
Dimensions	MessageType ResultCode

Table 6-180 Diameterproxy_SNR_Rejected_By_Unknown_User

Field	Details
Metric Name	nudr_diameterproxy_request_rejected_unknown_user_total
Description	The total number of subscriber notification messages rejected by an unknown user
Type	Counter
Dimensions	Not Applicable

Table 6-181 Diameterproxy_SNR_Rejected_By_Invalid_Service_Indication

Field	Details
Metric Name	nudr_diameterproxy_rejected_invalid_service_indication_requests_total{MessageType="SNR"}
Description	The total number of subscriber notification requests rejected by passing invalid service indications
Type	Counter
Dimensions	MessageType

Table 6-182 Diameterproxy_PNR_Messages

Field	Details
Metric Name	udr_diameterproxy_requests_total{MessageType="PNR"}
Description	The total number of Push notification messages sent from diameterproxy.
Type	Counter
Dimensions	MessageType

Table 6-183 Diameterproxy_PNA_Success_Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="PNA", ResultCode="2001"}
Description	The total number of success Push notification sent from diameterproxy.
Type	Counter
Dimensions	MessageType ResultCode

Table 6-184 Diameterproxy_PNA_Failure_Messages

Field	Details
Metric Name	nudr_diameterproxy_responses_total{MessageType="PNA", ResultCode != "2001"}
Description	The total number of failure push notification sent from diameterproxy.
Type	Counter
Dimensions	MessageType ResultCode

Table 6-185 Diameter_Total_Rest_request_recieved_From_Notify-service

Field	Details
Metric Name	nudr_diameterproxy_total_notification_received_total
Description	The total number of rest requests received from notify-service.
Type	Counter
Dimensions	Not Applicable

Table 6-186 Diameter_Total_Rest_Failure_requests_From_notify-service

Field	Details
Metric Name	nudr_diameterproxy_notification_failed_total
Description	The total number of rest requests are failed to send Push notification from the diameterproxy service.
Type	Counter
Dimensions	Not Applicable

Table 6-187 nudr_diameterproxy_config_service_update_total

Field	Details
Metric Name	nudr_diameterproxy_config_service_update_total{StatusCode="200/400"}
Description	The total number of update requests(PUT) along with successfully updated the data in microservice
Type	Counter
Dimensions	StatusCode

Table 6-188 diameter_request_processing_time_seconds

Field	Details
Metric Name	diameter_request_processing_time_seconds
Description	Average time for processing diameter messages and database operation.
Type	Timer
Dimensions	quantile InstanceIdentifier

Table 6-189 diameter_db_processing_time_seconds

Field	Details
Metric Name	diameter_db_processing_time_seconds
Description	Average time for only database operation for diameter messages.
Type	Timer
Dimensions	quantile InstanceIdentifier

Table 6-190 eir_default_response_imei_not_found

Field	Details
Metric Name	eir_default_response_imei_not_found
Description	Provides the number of ECA messages based on Interface (S13 or N17) with defaultResponses when IMEI is not present in EIR database.
Type	Counter
Dimensions	Interface

Table 6-191 eir_txn_default_response_sent

Field	Details
Metric Name	eir_txn_default_response_sent
Description	Provides the number of ECA messages based on Interface (S13 or N17) with defaultResponses when user name or IMSI is not present in the EIR database but IMEI is present in the EIR database.
Type	Counter
Dimensions	Interface

6.7 nudr-config-service Metrics

The following table provides metrics for nudr-config microservice of UDR.

Table 6-192 config_schema_validation_total

Field	Details
Metric Name	nuds_config_schema_validation_total{StatusCode="200/400"}
Description	Pegging schema validation with 200/400 as response code.
Type	Counter
Dimensions	StatusCode

Table 6-193 config_total_requests_total

Field	Details
Metric Name	nuds_config_total_requests_total{Method='GET/POST/PUT'}
Description	Pegging all the GET request received by configuration service
Type	Counter
Dimensions	Method

Table 6-194 config_total_responses

Field	Details
Metric Name	nuds_config_total_responses_total{Method='GET/PUT/POST',StatusCode="200/201/400/404/405/500"}
Description	Pegging the response of GET request with response code: 200/201/400/404/405/500
Type	Counter
Dimensions	Method StatusCode

Table 6-195 export_total_requests

Field	Details
Metric Name	nuds_config_export_total_requests_total{Method='POST'}
Description	Pegging all the POST request received by EXPORT API of configuration service
Type	Counter
Dimensions	Method

Table 6-196 export_total_responses

Field	Details
Metric Name	nuds_config_export_total_responses_total{Method='POST'}
Description	Pegging the response of EXPORT POST request with response code: 201/400/500
Type	Counter

Table 6-196 (Cont.) export_total_responses

Field	Details
Dimensions	Method StatusCode

Table 6-197 export_total_responses

Field	Details
Metric Name	nudr_config_export_status_total_requests_total{Method='GET'}
Description	Pegging all the GET request received by EXPORT status API of configuration service
Type	Counter
Dimensions	Method

Table 6-198 export_status_total_responses

Field	Details
Metric Name	nudr_config_export_status_total_responses_total{Method='GET'}
Description	Pegging the response of GET EXPORT status request with response code: 200/404/500
Type	Counter
Dimensions	Method StatusCode

Table 6-199 export_report_total_requests

Field	Details
Metric Name	nudr_config_export_report_total_requests_total{Method='GET'}
Description	Pegging all the GET Request received by EXPORT report API of config service
Type	Counter
Dimensions	Method

Table 6-200 export_report_total_responses

Field	Details
Metric Name	nudr_config_export_report_total_responses_total{Method='GET'}
Description	Pegging the response of GET EXPORT report request with response code: 200/404/500
Type	Counter
Dimensions	Method StatusCode

Table 6-201 export_download_total_requests

Field	Details
Metric Name	nudr_config_export_download_total_requests_total{Method='GET'}
Description	Pegging all the GET request received by EXPORT download API of configuration service

Table 6-201 (Cont.) export_download_total_requests

Field	Details
Type	Counter
Dimensions	Method

Table 6-202 export_download_total_responses

Field	Details
Metric Name	nldr_config_export_download_total_responses_total{Method='GET'}
Description	Pegging the response of GET EXPORT download request with response code: 200/404/500
Type	Counter
Dimensions	Method StatusCode

Table 6-203 import_total_requests

Field	Details
Metric Name	nldr_config_import_total_requests_total{Method='POST'}
Description	Pegging all the POST request received by IMPORT API of configuration service
Type	Counter
Dimensions	Method

Table 6-204 import_total_responses

Field	Details
Metric Name	nldr_config_import_total_responses_total{Method='POST'}
Description	Pegging the response of IMPORT POST request with response code: 201/400/500
Type	Counter
Dimensions	Method StatusCode

Table 6-205 import_status_total_requests

Field	Details
Metric Name	nldr_config_import_status_total_requests_total{Method='GET'}
Description	Pegging all the GET request received by IMPORT status API of configuration service
Type	Counter
Dimensions	Method

Table 6-206 import_status_total_responses

Field	Details
Metric Name	nldr_config_import_status_total_responses_total{Method='GET'}
Description	Pegging the response of GET IMPORT status request with response code: 200/404/500

Table 6-206 (Cont.) import_status_total_responses

Field	Details
Type	Counter
Dimensions	Method StatusCode

Table 6-207 import_report_total_requests

Field	Details
Metric Name	nudr_config_import_report_total_requests_total{Method='GET'}
Description	Pegging all the GET requests received by IMPORT report API of configuration service
Type	Counter
Dimensions	Method

Table 6-208 import_report_total_requests

Field	Details
Metric Name	nudr_config_import_report_total_responses_total{Method='GET'}
Description	Pegging the response of GET IMPORT report request with response code: 200/404/500
Type	Counter
Dimensions	Method StatusCode

Table 6-209 export_tool_controlled_shutdown

Field	Details
Metric Name	export_tool_controlled_shutdown
Description	This metric indicates the operational status. For example, 0 - NORMAL 1 - COMPLETE_SHUTDOWN
Type	Gauge
Dimensions	NA

Table 6-210 operational_status

Field	Details
Metric Name	nudr_config_operational_status
Description	This metric indicates the operational status of UDR. For example, 0 - NORMAL 1 - COMPLETE_SHUTDOWN
Type	Gauge
Dimensions	NA

6.8 nrf-client Metrics

The following table provides metrics for nrf-client.

Table 6-211 NRF Instance Status

Field	Details
Metric Name	nrfclient_nrf_operative_status
Description	The current operative status of the NRF Instance. 0 - NRF is unavailable/unhealthy 1- NRF is available/healthy
Type	Gauge
Dimensions	NrfUri - URI of the NRF Instance

Table 6-212 NRF Instance Status Count

Field	Details
Metric Name	nrfclient_nrf_status_total
Description	Total number of times a NRF Instance is marked as Healthy or Un-healthy apiRoot shall be specified in the format 'scheme':'fqdn':'port' If Health Check procedure is disabled, all NRF Instances are marked as HEALTHY after Successful NfRegistration.
Type	Counter
Dimensions	NrfUri- URI of the NRF InstanceHealthStatusFailureReason - Reason for the status

Table 6-213 NRF Instance Consecutive Healthy Count

Field	Details
Metric Name	nrfclient_nrf_successive_healthy_count
Description	The metric shows the consecutive number of times the NRF is considered as healthy. The metric shall have a minimum value of 0 and maximum value of healthCheckCount.
Type	Counter
Dimensions	NrfUri- URI of the NRF Instance

Table 6-214 NRF Instance Consecutive Unhealthy Count

Field	Details
Metric Name	nrfclient_nrf_successive_unhealthy_count
Description	The metric shows the consecutive number of times the NRF is considered as healthy. The metric shall have a minimum value of 0 and maximum value of healthCheckCount.
Type	Counter
Dimensions	NrfUri- URI of the NRF Instance

Table 6-215 DNS lookup requests

Field	Details
Metric Name	nrfclient_dns_lookup_request_total
Description	Total number of times a DNS lookup request is sent to the alternate route service. This metric replaces ocpm_ar_lookup_request_total.
Type	Scheme - http or httpsVirtualFqdn - Fqdn that shall be used by the alternate service for the DNS lookup
Dimensions	The metric shall be pegged only if enableVirtualNrfResolution is set to true

Table 6-216 DNS lookup requests

Field	Details
Metric Name	nrfclient_dns_lookup_response_total
Description	Total number of times a DNS lookup response is received from the alternate route service. This metric replaces ocpm_ar_lookup_response_total.
Type	Scheme - HTTP or HTTPS VirtualFqdn - Fqdn that shall be used by the alternate service for the DNS lookup HttpStatusCode - The status code as received in the response.
Dimensions	The metric shall be pegged only if enableVirtualNrfResolution is set to true

Table 6-217 DNS setup requests

Field	Details
Metric Name	nrfclient_dns_register_request_total
Description	Total number of times a DNS register request is sent to the alternate route service. This metric replaces ocpm_ar_setup_total.
Type	Scheme - http or https VirtualFqdn - FQDN that shall be used by the alternate service for the DNS lookup
Dimensions	The metric shall be pegged only if enableVirtualNrfResolution is set to true.

Table 6-218 NF Load

Field	Details
Metric Name	nrfclient_perf_info_nf_profile_load
Description	The current Load of the NF
Type	Gauge
Dimensions	Not Applicable

Table 6-219 Current NF Status

Field	Details
Metric Name	nrfclient_current_nf_status
Description	The current operative status of the NF. The gauge indicates the status as below: 0 - REGISTERED1 - DEREGISTERED2 - SUSPENDED3 - UNDISCOVERABLE4 - UNKNOWN

Table 6-219 (Cont.) Current NF Status

Field	Details
Type	Gauge
Dimensions	NfType - The NF's NfType as present in the registered NfProfile. NfInstanceId - The NF's nfInstanceId as present in the registered NfProfile. NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 6-220 Current NF Status with NRF

Field	Details
Metric Name	nrfclient_nf_status_with_nrf
Description	The operative status of the NF communicated to the NRF. The gauge must indicate the status as follows: • 0 - REGISTERED • 1 - DEREGISTERED • 2 - SUSPENDED • 3 - UNDISCOVERABLE • 4 - UNKNOWN
Type	Gauge
Dimensions	NfType - The NF's NfType as present in the registered NfProfile. NfInstanceId - The NF's nfInstanceId as present in the registered NfProfile. NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 6-221 Health Status Subscription Requests

Field	Details
Metric Name	nrfclient_conn_in_subscribe_nrf_health_status_total
Description	Total number of Subscribe requests from NF service to to NRF-client to monitor the health status of the configured NRFs
Type	counter
Dimensions	NotificationURI - The URI received in the request

Table 6-222 Health Status Subscription Responses

Field	Details
Metric Name	nrfclient_conn_out_subscribe_nrf_health_status_total
Description	Total number of Subscribe response from NRF-Client to the subscribed NF service
Type	counter
Dimensions	NotificationURI - The URI received in the request HttpStatusCode - The HttpStatusCode of the response

Table 6-223 Health Status Notification requests

Field	Details
Metric Name	nrfclient_conn_out_notify_nrf_health_status_total

Table 6-223 (Cont.) Health Status Notification requests

Field	Details
Description	Total number of notification sent to subscribed NF services due to health status update
Type	counter
Dimensions	NotificationURI - The URI to which notifications are sent.

Table 6-224 Health Status Notification responses

Field	Details
Metric Name	nrfclient_conn_in_notify_nrf_health_status_total
Description	Total number of response received for notification request from subscriber
Type	counter
Dimensions	NotificationURI - The URI to which notification is sent. , HttpStatusCode - The HttpStatusCode as sent by the subscriber,

Table 6-225 Requests sent by NRF-Client to NRF

Field	Details
Metric Name	nrfclient_nw_conn_out_request_total
Description	Total number of times NRF-client has sent a request to NRF. This includes autonomous requests as well as on-demand requests.
Type	counter
Dimensions	MessageType - The service request Type. NfType - The NF's NfType as present in the registered NfProfile. NfInstanceId - The NF's nfInstanceId as present in the registered NfProfile. NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 6-226 Responses Received by NRF-Client from NRF

Field	Details
Metric Name	nrfclient_nw_conn_in_response_total
Description	Total number of times NRF-client has received a response from NRF.
Type	counter
Dimensions	MessageType - The service request Type. NfType - The NF's NfType as present in the registered NfProfile. NfInstanceId - The NF's nfInstanceId as present in the registered NfProfile. NfFqdn - The NF's FQDN as present in the registered NfProfile. StatusCode - The HttpStatusCode as received from NRF.

Table 6-227 Notification Requests received at NRF-Client

Field	Details
Metric Name	nrfclient_nw_conn_in_notify_request_total
Description	Total number of nfStatusNotify requests received from NRF

Table 6-227 (Cont.) Notification Requests received at NRF-Client

Field	Details
Type	counter
Dimensions	EventType - The EventType for which the notification is triggered, NfInstanceId - The NfInstanceId for which the notification is triggered.

Table 6-228 Notification responses sent by NRF-Client

Field	Details
Metric Name	nrfclient_nw_conn_out_notify_response_total
Description	Total number of nfStatusNotify responses sent to NRF
Type	counter
Dimensions	EventType - The EventType for which the notification is triggered, NfInstanceId - The NfInstanceId for which the notification is triggered. HttpStatusCode - The HttpStatusCode sent by NRF-Client.

Table 6-229 Processing Time for Messages Between NRF-client and NRF

Field	Details
Metric Name	nrfclient_network_message_processing_latency
Description	Total message processing time duration
Type	Histogram
Dimensions	MessageType - The service request Type. NfType - The NF's NfType as present in the registered NfProfile. NfInstanceId - The NF's nfInstanceId as present in the registered NfProfile. NfFqdn - The NF's FQDN as present in the registered NfProfile.

Table 6-230 Requests Sent to app-info service

Field	Details
Metric Name	nrfclient_app_info_conn_out_request_total
Description	Total Number of requests sent to app-info service.
Type	Histogram
Dimensions	NfType - The NF's NfType as present in the registered NfProfile.

Table 6-231 Responses Received from app-info service

Field	Details
Metric Name	nrfclient_app_info_conn_in_response_total
Description	Total number of responses received from app-info service
Type	Counter
Dimensions	NfType - The NF's NfType as present in the registered NfProfile. StatusCode - The HttpStatusCode sent by app-info.

6.9 nudr-migration-service Metrics

The following table provides metrics for nudr-migration microservice of UDR.

Table 6-232 Total Records Start/End Range

Field	Details
Metric Name	nudr_migration_total_requests_total
Description	Total number of requests with start and end range
Type	Counter
Dimensions	Not Applicable

Table 6-233 Total Number of Requests Read

Field	Details
Metric Name	nudr_migration_total_requests_read_total
Description	Total number of requests read from 4G UDR
Type	Counter
Dimensions	Not Applicable

Table 6-234 Total Number of Requests Processed

Field	Details
Metric Name	nudr_migration_total_requests_processed_total
Description	Total number of requests processed to 5G UDR
Type	Counter
Dimensions	Not Applicable

Table 6-235 Total number of Requests Read with Success or Failure

Field	Details
Metric Name	nudr_migration_total_requests_read_status_total{Status="success/failure"}
Description	Total number of requests read from 4G UDR with success/failure
Type	Counter
Dimensions	Status

Table 6-236 Total number of Requests Read with Success or Failure

Field	Details
Metric Name	nudr_migration_total_requests_processed_status_total{Status="success/failure"}
Description	Total number of requests processed to 5G UDR with success/failure
Type	Counter
Dimensions	Status

Table 6-237 Total Number of Subscribers Not Found

Field	Details
Metric Name	nudr_migration_total_no_of_source_subscriber_not_exist_total
Description	Total number of subscribers not found while reading from 4G UDR
Type	Counter
Dimensions	Not Applicable

Table 6-238 Total Number of Target Keys

Field	Details
Metric Name	nudr_migration_total_no_of_target_udr_keys_exist_total
Description	Total number of target keys exists in 5G UDR
Type	Counter
Dimensions	Not Applicable

Table 6-239 Config Service Update Total

Field	Details
Metric Name	nudr_migration_tool_config_service_update_total
Description	Total number of updates made using nudr_migration configuration API
Type	Counter
Dimensions	Not Applicable

6.10 nudr-bulk-import-service Metrics

The following table provides metrics for nudr-bulk-import microservice of UDR.

Table 6-240 nudr_bulk_import_records_provisioned_total

Field	Details
Metric Name	nudr_bulk_import_records_provisioned_total{StatusCode="204/201/500/400/404/503",Status="Success/Failure", NFType="PCF/SLF/PCF-VSA/EIR"}
Description	Total number of success or failure records provisioned in UDR for PCF or SLF or PCF-VSA or EIR
Type	Counter
Dimensions	StatusCode Status NFType

Table 6-241 nudr_bulk_import_csvfile_records_read_total

Field	Details
Metric Name	nudr_bulk_import_csvfile_records_read_total{Method="DELETE/PUT/POST/PATCH", Status="Success/Failure"}

Table 6-241 (Cont.) nldr_bulk_import_csvfile_records_read_total

Field	Details
Description	Total number of successful record reads from CSV file for DELETE, PUT, PATCH, POST operation.
Type	Counter
Dimensions	Method Status

Table 6-242 nldr_bulk_import_records_processed_total

Field	Details
Metric Name	nldr_bulk_import_records_processed_total{Method = "POST/PUT/DELETE/PATCH", StatusCode="201/204/500/404/400/503", Status="Success/Failure"}
Description	Total number of records processed from CSV file.
Type	Counter
Dimensions	Method StatusCode Status

Table 6-243 nldr_bulk_import_tool_pvc_full_usage

Field	Details
Metric Name	nldr_bulk_import_tool_pvc_full_usage{container="nldr-bulk-import",namespace="ocudr"}
Description	Storage for Bulk Import tool is full
Type	Counter
Dimensions	container="nldr-bulk-import"

Table 6-244 nldr_xmltocsx_xmlfile_records_read_total

Field	Details
Metric Name	nldr_xmltocsx_xmlfile_records_read_total{Status="Success/Failure"}
Description	Total number of successful record reads from ixml file.
Type	Counter
Dimensions	Status

Table 6-245 nldr_xmltocsx_records_processed_total

Field	Details
Metric Name	nldr_xmltocsx_records_processed_total{Method = "POST/PUT/DELETE/PATCH", Status="Success/Failure"}
Description	Total number of xml records converted Success/Failure
Type	Counter
Dimensions	Method Status

Table 6-246 nldr_bulk_import_tool_pvc_full_usage

Field	Details
Metric Name	nldr_bulk_import_tool_pvc_full_usage{container="nldr-xmltocsv",namespace="ocldr"}
Description	Storage for XML Bulk Import tool is full
Type	Counter
Dimensions	container="nldr-xmltocsv"

Table 6-247 nldr_pdbi_to_csv_file_records_read

Field	Details
Metric Name	nldr_pdbi_to_csv_file_records_read{Status="Success"}
Description	Total number of successful record reads from PDBI file
Type	Counter
Dimensions	Status

Table 6-248 nldr_pdbi_to_csv_records_processed

Field	Details
Metric Name	nldr_pdbi_to_csv_records_processed(Status="Success",Method="CREATE,MODIFY,DELETE")
Description	Total number of pdbi records processed
Type	Counter
Dimensions	Method Status

Table 6-249 nldr_pdbi_to_csv_records_processed

Field	Details
Metric Name	nldr_pdbi_to_csv_records_processed_failed_total
Description	Total number of pdbi records processed failed
Type	Counter
Dimensions	container="nldr-pdbitocsv"

Table 6-250 nldr_bulk_import_tool_pvc_full_usage

Field	Details
Metric Name	nldr_bulk_import_tool_pvc_full_usage{container="nldr-pdbitocsv",namespace="ocldr"}
Description	Storage for PDBI Bulk Import tool is full
Type	Counter
Dimensions	container="nldr-bulk-import"

Table 6-251 bulkimport_files_fetched_from_remote_total

Field	Details
Metric Name	bulkimport_files_fetched_from_remote_total
Description	Total number of files that are transferred from remote to pvc
Type	Counter
Dimensions	NA

Table 6-252 bulkimport_files_fetched_from_remote_status_total

Field	Details
Metric Name	bulkimport_files_fetched_from_remote_status_total{status="success/failed",reason="CANNOT_CONNECT_TO_HOST/ACCESS_DENIED_TO_REMOTE/FILE_SIZE_NOT_AS_EXPECTED/LOGIN_DENIED/OUT_OF_SPACE"}
Description	Total number of files that are transferred from remote to pvc with status success/failure
Type	Counter
Dimensions	Status

Table 6-253 bulkimport_files_transferred_to_remote_total

Field	Details
Metric Name	bulkimport_files_transferred_to_remote_total
Description	Total number of files that are transferred from pvc to remote
Type	Counter
Dimensions	NA

Table 6-254 bulkimport_files_transferred_to_remote_status_total

Field	Details
Metric Name	bulkimport_files_transferred_to_remote_status_total{status="success/failed",reason="CANNOT_CONNECT_TO_HOST/ACCESS_DENIED_TO_REMOTE/FILE_SIZE_NOT_AS_EXPECTED/LOGIN_DENIED/OUT_OF_SPACE"}
Description	Total number of files that are transferred from pvc to remote with status success/failure
Type	Counter
Dimensions	status

Table 6-255 bulkimport_transfer_in_status

Field	Details
Metric Name	bulkimport_transfer_in_status
Description	Transfer-in status , 0 - success and 1- failure
Type	Gauge
Dimensions	NA

Table 6-256 bulkimport_transfer_out_status

Field	Details
Metric Name	bulkimport_transfer_out_status
Description	Transfer-in status , 0 - success and 1- failure
Type	Gauge
Dimensions	NA

6.11 nudr-export-tool service Metrics

The following table provides metrics for nudr-export-tool microservice of UDR.

① Note

Prometheus automatically appends the word total to all the counter type metrics that are displayed on the Prometheus dashboard.

Table 6-257 Export Status Aggregate

Field	Details
Metric Name	export_tool_status_total{status="success/fail"}
Description	Total number of success or failure export operations.
Type	Counter
Dimensions	Status

Table 6-258 Total Export Operations

Field	Details
Metric Name	export_tool_backUp_taken_total
Description	Total number of export operation triggered.
Type	Counter
Dimensions	NA

Table 6-259 Export Failure Indicator

Field	Details
Metric Name	export_failure
Description	Gauge indicating export operation failure. Values: 0: Success 1: Failure
Type	Gauge
Dimensions	NA

Table 6-260 PVC Full Indication

Field	Details
Metric Name	export_tool_full_usage
Description	Gauge indicating PVC full usage. Values: PVC Full: 1 PVC normal: 0
Type	Gauge
Dimensions	NA

Table 6-261 Total File Transfer Attempts

Field	Details
Metric Name	nldr_export_tool_total_file_transfers_total
Description	Total number of file transfer attempts.
Type	Counter
Dimensions	NA

Table 6-262 Successfull File Transfer Attempts

Field	Details
Metric Name	nldr_export_tool_successful_file_transfers_total
Description	Total number of successful file transfer attempts
Type	Counter
Dimensions	status

Table 6-263 Failed File Transfer Attempts

Field	Details
Metric Name	nldr_export_tool_failed_file_transfers_total
Description	Total number of failed file transfer attempts. status=failed,reason=CANNOT_CONNECT_TO_HOST,ACCESS_DENIED_TO_REMOTE,FILE_SIZE_NOT_AS_EXPECTED, LOGIN_DENIED,OUT_OF_SPACE
Type	Counter
Dimensions	status, reason

Table 6-264 Operational state of the NF

Field	Details
Metric Name	export_tool_controlled_shutdown
Description	Gives the operational state of the NF. For example, 0- NORMAL and 1-COMplete_shutdown
Type	Gauge
Dimensions	NA

6.12 nudr-ondemand-migration-service Metrics

The following table provides metrics for nudr-ondemand-migration microservice of UDR.

Table 6-265 ondemand_migration_requests_total

Field	Details
Metric Name	nudr_ondemand_migration_requests_total
Description	Total number of request received from dr-service for migration 4G subscriber in 5G UDR.
Type	Counter
Dimensions	Not Applicable

Table 6-266 ondemand_migration_requests_read_total

Field	Details
Metric Name	nudr_ondemand_migration_requests_read_total
Description	Total number of sent request to target UDR for read subscriber details.
Type	Counter
Dimensions	Not Applicable

Table 6-267 ondemand_migration_requests_read_status_total

Field	Details
Metric Name	nudr_ondemand_migration_requests_read_status_total{Status:success/failure}
Description	The total success/failure read messages sent target UDR.
Type	Counter
Dimensions	Status

Table 6-268 ondemand_migration_requests_processed_total

Field	Details
Metric Name	nudr_ondemand_migration_requests_processed_total
Description	The total processed messages coming from target UDR.
Type	Counter
Dimensions	Not Applicable

Table 6-269 ondemand_migration_requests_processed_status_total

Field	Details
Metric Name	nudr_ondemand_migration_requests_processed_status_total{Status:success/failure}
Description	Total success/failure processed messages coming from 4G UDR.
Type	Counter
Dimensions	Status

Table 6-270 `ondemand_migration_subscriber_not_exist_source_total`

Field	Details
Metric Name	<code>nudr_ondemand_migration_subscriber_not_exist_source_total</code>
Description	Total subscriber not found from target UDR.
Type	Counter
Dimensions	Not Applicable

Table 6-271 `ondemand_migration_delete_subscriber_source_total`

Field	Details
Metric Name	<code>nudr_ondemand_migration_subscriber_not_exist_source_total</code>
Description	Total deleted subscriber from target UDR when delete subscriber flag is enabled.
Type	Counter
Dimensions	Not Applicable

Table 6-272 `ondemand_migration_config_service_update_total`

Field	Details
Metric Name	<code>nudr_ondemand_migration_config_service_update_total{StatusCode="200/400"}</code>
Description	Total success(200)/failure(400) update requests from CNC Console.
Type	Counter
Dimensions	StatusCode

6.13 app-info Metrics

The following table provides metrics for app-info microservice.

Table 6-273 `appinfo_service_running`

Field	Details
Metric Name	<code>appinfo_service_running</code>
Description	The status of monitored services
Type	Integer: 0,1
Dimensions	- service - namespace - category

Table 6-274 `appinfo_category_running`

Field	Details
Metric Name	<code>appinfo_category_running</code>
Description	The status of monitored categories
Type	Integer: 0,1
Dimensions	- namespace - category

Table 6-275 appinfo_category_good

Field	Details
Metric Name	appinfo_category_good
Description	The readiness of monitored categories
Type	Integer: 0,1
Dimensions	- namespace - category

6.14 perf-info Metrics

The following table provides metrics for perf-info microservice.

Table 6-276 nf_load_info

Field	Details
Metric Name	nf_load_info
Description	Service load info
Type	Float
Dimensions	- service - namespace

Table 6-277 jvm_cpu_usage

Field	Details
Metric Name	jvm_cpu_usage
Description	Springboot per service jvm_cpu_usage
Type	Float
Dimensions	- service - namespace

Table 6-278 jvm_memory

Field	Details
Metric Name	jvm_memory
Description	Springboot per service jvm_memory
Type	Float
Dimensions	- service - namespace

Table 6-279 cgroup_cpu_nanoseconds

Field	Details
Metric Name	cgroup_cpu_nanoseconds
Description	It reports the total CPU time in nanoseconds on each CPU core for all tasks in the cgroup
Type	Float

Table 6-279 (Cont.) cgroup_cpu_nanoseconds

Field	Details
Dimensions	NA

Table 6-280 cgroup_memory_bytes

Field	Details
Metric Name	cgroup_memory_bytes
Description	It reports the memory usage which is fairly straight forward to monitor
Type	Float
Dimensions	NA

Table 6-281 load_level

Field	Details
Metric Name	load_level
Description	The overload manager load level info
Type	Float
Dimensions	0,1,2,3,4

Table 6-282 service_resource_stress

Field	Details
Metric Name	service_resource_stress
Description	Tracks CPU, memory, failures, and pending requests based on which the service calculates the overload level.
Type	NA
Dimensions	- service - type

Table 6-283 service_resource_overload_level

Field	Details
Metric Name	service_resource_overload_level
Description	Tracks individual resource overload level.
Type	0,1,2,3,4
Dimensions	- service - type

6.15 Diameter Gateway Metrics

The following table provides metrics for Diameter Gateway.

Table 6-284 diam_conn_network

Field	Details
Metric Name	diam_conn_network
Description	Tracks total number of active diameter connections with network peers.
Type	Gauge
Dimensions	destHost, destRealm, origHost, origRealm, direction from CER

Table 6-285 diam_conn_network

Field	Details
Metric Name	diam_conn_network_uptime_ms
Description	Tracks duration of active diameter connection with network peer.
Type	Histogram
Dimensions	destHost, destRealm, origHost, origRealm, direction, sduration

Table 6-286 diam_conn_local

Field	Details
Metric Name	diam_conn_local
Description	Tracks active diameter connection with local backend services.
Type	Gauge
Dimensions	destHost, destRealm, origHost, origRealm, direction from CER

Table 6-287 diam_conn_local_uptime_ms

Field	Details
Metric Name	diam_conn_local_uptime_ms
Description	Tracks overall up time of a diameter connection with local services.
Type	Histogram
Dimensions	destHost, destRealm, origHost, origRealm, direction, sduration

Table 6-288 diam_request_network_total

Field	Details
Metric Name	diam_request_network_total
Description	Tracks total number of request messages of a given command code to the network or from the network.
Type	Counter
Dimensions	destHost, destRealm, origHost, origRealm, appId, cmdCode, msgType, direction

Table 6-289 diam_response_network_total

Field	Details
Metric Name	diam_response_network_total

Table 6-289 (Cont.) `diam_response_network_total`

Field	Details
Description	Tracks total number of answer messages of a given command code to the network or from the network.
Type	Counter
Dimensions	reqDestHost, reqDestRealm, reqOrigHost, reqOrigRealm, applId, cmdCode, <i>msgType</i> , direction, responseCode

Table 6-290 `diam_response_network_processing_latency_ms`

Field	Details
Metric Name	<code>diam_response_network_processing_latency_ms</code>
Description	Tracks overall response time of a diameter answer of a given command code to or from the network.
Type	Histogram
Dimensions	reqDestHost, reqDestRealm, reqOrigHost, reqOrigRealm, applId, cmdCode, <i>msgType</i> , direction, duration

Table 6-291 `diam_request_local_total`

Field	Details
Metric Name	<code>diam_request_local_total</code>
Description	Tracks total number of request messages of a given command code to or from local service.
Type	Counter
Dimensions	destHost, destRealm, origHost, origRealm, applId, cmdCode, <i>msgType</i> , direction

Table 6-292 `diam_response_local_total`

Field	Details
Metric Name	<code>diam_response_local_total</code>
Description	Tracks total number of answer messages of a given command code to or from the local service.
Type	Counter
Dimensions	reqDestHost, reqDestRealm, reqOrigHost, reqOrigRealm, applId, cmdCode, <i>msgType</i> , direction, responseCode

Table 6-293 `diam_response_local_processing_latency_ms`

Field	Details
Metric Name	<code>diam_response_local_processing_latency_ms</code>
Description	Tracks overall response time of a diameter answer of a given command code to or from local service.
Type	Histogram
Dimensions	reqDestHost, reqDestRealm, reqOrigHost, reqOrigRealm, applId, cmdCode, <i>msgType</i> , direction, duration

Table 6-294 diam_request_inter_total

Field	Details
Metric Name	diam_request_inter_total
Description	Tracks total number of request messages sent or received during inter-pod routing.
Type	Counter
Dimensions	destHost, destRealm, origHost, origRealm, appId, cmdCode, <i>msgType</i> , direction, srcPodIp, dsrPodIp

Table 6-295 diam_response_inter_total

Field	Details
Metric Name	diam_response_inter_total
Description	Tracks total number of answer messages sent or received during inter-pod routing.
Type	Counter
Dimensions	reqDestHost, reqDestRealm, reqOrigHost, reqOrigRealm, appId, cmdCode, <i>msgType</i> , direction, srcPodIp, dsrPodIp, responseCode

Table 6-296 diam_response_inter_latency_ms

Field	Details
Metric Name	diam_response_inter_latency_ms
Description	Tracks overall response time of a diameter answer of a given command code for inter-pod messages. It tracks response only from a pod that initiates inter-pod routing.
Type	Histogram
Dimensions	reqDestHost, reqDestRealm, reqOrigHost, reqOrigRealm, appId, cmdCode, <i>msgType</i> , direction, srcPodIp, dsrPodIp, duration

Table 6-297 ocudr_pod_congestion_state

Field	Details
Metric Name	ocudr_pod_congestion_state
Description	Overall diameter gateway pod state. Different states of pod: 0 - NORMAL state 1- DOC state (Danger of Congestion) 2- CONGESTED state
Type	Gauge
Dimensions	NA

Table 6-298 ocudr_pod_resource_congestion_state

Field	Details
Metric Name	ocudr_pod_resource_congestion_state

Table 6-298 (Cont.) ocudr_pod_resource_congestion_state

Field	Details
Description	Different resource state of the pod. Different types of resource in pod state are as follows: • cpu • memory • queueSize Each resource state as follows: • 0 - NORMAL state • 1- DOC state (Danger of Congestion) • 2- CONGESTED state
Type	Gauge
Dimensions	type

Table 6-299 ocudr_diam_congestion_message_reject_total

Field	Details
Metric Name	ocudr_diam_congestion_message_reject_total
Description	Rejected messages if the configured priority is less than message priority.
Type	Counter
Dimensions	priority,responseCode,destHost,destRealm,origRealm,origHost,appld,cmdCode,msgType

6.16 Nudr-dbcdr-auditor-service

The following table provides metrics for Auditor Service.

Table 6-300 nudr_dbcr_auditor_service_monitor_total

Field	Details
Metric Name	nudr_dbcr_auditor_service_monitor_total
Description	The total number of completed exception tables audit cycle.
Type	Counter
Dimensions	NA

Table 6-301 nudr_dbcr_auditor_service_clear_total

Field	Details
Metric Name	nudr_dbcr_auditor_service_clear_total
Description	The total number of completed exception tables clearing cycle.
Type	Counter
Dimensions	NA

Table 6-302 nudr_dbcrc_auditor_service_delete_total

Field	Details
Metric Name	nudr_dbcrc_auditor_service_delete_total
Description	The total number of deletions done from data and exception tables which includes total number of rows deleted from IDXTODATA/IDXTODATA\$EX, other data tables and corresponding exception tables.
Type	Counter
Dimensions	NA

A

Rollback Instructions for PCF Data to Release v15

In this Appendix, you will learn to revert to 29.519 v15.3.0 for PCF Data.

Note

In a deployed UDR, you must 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. If there are multiple versions, it will create inconsistency in the json data stored on UDR.

Note

If the user wants to be compatible with 29.519 v15.3 of PCF data, follow the instructions while deploying UDR and do not change, if there are some subscribers provisioned using any older schema.

Schema Rollback to 29.519 15.3.0

A script (***rollbackPCFschema_15_3.py***) is provided for this procedure (as part of the customer documentation). The steps to run the script on the CNE bastion are:

1. Open the script and edit the following details as per the udrdb configurations.

```
mydb = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    passwd="xxxxxxx",  
    database="udrdb")
```

2. Run the following command:

```
python rollbackPCFschema_15_3.py
```

3. If python cannot be installed on the customer setup, then login to one of the SQL node pods. Run the SQL commands in same order as shared in the python script.

B

Error Code Catalogue

The following table provides error code details:

Table B-1 Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Header Related Issues	-	-
Error Code: E001 HTTP Response Code: 415 Category: HDRVLD Error Description: Unsupported media type header	Content-Type header is incorrect.	Modify Content-Type to application/json or application/patch-json based on the operation type (PUT, POST, or PATCH).
Error Code: E002 HTTP Response Code: 406 Category: HDRVLD Error Description: Media type header not acceptable.	Accept header is incorrect.	Modify accept header to application/json or application/patch-json based on the operation type (PUT, POST or PATCH).
Error Code: E003 HTTP Response Code: 412 Category: HDRVLD Error Description: ETag header out of sync	ETag header value is not in sync with UDR.	Perform a GET operation to get the latest ETag and send the update operation with updated ETag.
Error Code: E004 HTTP Response Code: 409 Category: HDRVLD Error Description: Conflict in values or sequence number out of sync	Sequence number header value is not in sync with UDR.	Perform a UDR operation to get the latest sequence number and send the PUR operation with updated sequence number.
Error Code: E005 to E020 HTTP Response Code: Valid HTTP status code Category: HDRVLD Error Description: Reserved for generic header related issues	NA	NA
URI Related Issues:	-	-
Error Code: E021 HTTP Response Code: 400 Category: URIVLD Error Description: Invalid URI sent from client	The URI in the request is invalid or does not comply to the specification	Check the specification for the right URI. You can also refer to <i>Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide</i> to get the supported APIs.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E022 HTTP Response Code: 400 Category: URIVLD Error Description: Payload should contain key-value pair present in URI	The ueld in URI is not present in the payload for provisioning.	Make sure to include the ueld used in URI as part of the payload for CREATE operations.
Error Code: E023 HTTP Response Code: 400 Category: URIVLD Error Description: Failed to fetch jpath for given URI	The URI format is incorrect.	Make sure to send the correct URI, refer to <i>Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide</i> .
Error Code: E024 HTTP Response Code: 400 Category: URIVLD Error Description: Mandatory query parameter is missing	Mandatory query parameter required for the GET and DELETE operations are missing in the URI.	Check the <i>Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide</i> to get the mandatory query parameter to perform the operation.
Error Code: E025 HTTP Response Code: 400 Category: URIVLD Error Description: Mandatory query parameter is incorrect	Mandatory query parameter required for the GET and DELETE operations are incorrect.	Check the <i>Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide</i> to get the mandatory query parameter format to perform the operation.
Error Code: E026 HTTP Response Code: 400 Category: URIVLD Error Description: Unknown key, the key provided in the request is invalid	The subscriber key used with the URI ueld is not supported.	Check if one of the following key types is used in ueld:: • IMSI • MSISDN • NAI • AccountId • ExtId • IMEI (Supported only for EIR mode) • IMEISV (Supported only for EIR mode)
Error Code: E027 HTTP Response Code: 400 Category: URIVLD Error Description: Invalid key value	The subscriber key value used with the URI ueld is not supported.	Check if the key value format in ueld is sent correctly in the URI.
Error Code: E028 HTTP Response Code: 400 Category: URIVLD Error Description: Query parameters not expected	The URI has query parameters that are not expected for this operation.	Remove query parameters from the request URI.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E029 HTTP Response Code: 400 Category: URIVLD Error Description: Invalid query parameters supplied	The URI has query parameters that are not supported for this operation.	Remove query parameters which are not supported for the operation. Refer to <i>Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide</i> for the supported query parameters.
Error Code: E030 to E050 HTTP Response Code: Valid HTTP status code Category: URIVLD Error Description: Reserved for generic URI errors	NA	NA
Error Code: E051 to E060 HTTP Response Code: Valid HTTP status code Category: URIVLD Error Description: Reserved for SLF specific URI errors	NA	NA
Error Code: E061 HTTP Response Code: 400 Category: URIVLD Error Description: Keys other than IMEI or IMEISV in URI, not allowed for EIR profile update	Keys other than IMEI and IMEISV are used in URI for EIR mode.	Use IMEI or IMEISV only for ueld in URI for EIR operations.
Error Code: E062 to E070 HTTP Response Code: 400 Category: URIVLD Error Description: Reserved for EIR specific URI errors	NA	NA
Error Code: E071 HTTP Response Code: 400 Category: URIVLD Error Description: Invalid serving plmnId sent in URI	The UDM API used has a invalid plmnId in URI.	Make sure plmnid is sent correctly.
Error Code: E072 HTTP Response Code: 400 Category: URIVLD Error Description: Invalid pduSessionId sent in URI	The UDM API used has a invalid pduSessionId in URI.	Make sure pduSessionId is sent correctly.
Error Code: E073 HTTP Response Code: 400 Category: URIVLD Error Description: Query parameters ue-Id and one of nf-Instance-Id or delete-all-nfs are mandatory.	Mandatory query parameters required for the operation are not provided. ue-Id and nf-Instance-Id/delete-all-nfs are mandatory.	Make sure the mandatory query parameters are specified in the possible cause.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E074 to E080 HTTP Response Code: Valid HTTP status code Category: URIVLD Error Description: Reserved for UDM consumer URI errors	NA	NA
Error Code: E081 HTTP Response Code: 400 Category: URIVLD Error Description: Quota type in url and payload must be same	Quota-type parameter in the payload does not match the URL.	Provide the correct quota-type in payload and URI and make sure that they are the same.
Error Code: E082 HTTP Response Code: 400 Category: URIVLD Error Description: 4G attribute name must be same in the URL and payload.	4G-attribute-name parameter in the payload does not match with the one in the URL.	Provide the correct 4G-attribute-name in payload and URI and make sure that they are same.
Error Code: E083 to E100 HTTP Response Code: Valid HTTP status code Category: URIVLD Error Description: Reserved for PCF or VSA consumer URI related issues	NA	NA
Payload Related Issues	-	-
Error Code: E101 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid content request data supplied	- Request body in the request is invalid. - Format of the JSON data is incorrect. - Fields which are supposed to be json arrays are not in right format. - Profile-data is not present for CREATE operations. - op, filters, or patchFields are not present in VSA operations.	Fix the request payload if any of the possible causes mentioned occurs.
Error Code: E102 HTTP Response Code: 400 Category: REQVLD Error Description: Payload must not be empty	Request body is invalid. The payload is empty for a CREATE or UPDATE operations.	Send the request payload required as per the rest specification. Refer to <i>Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide</i> .

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E103 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid JSON payload	The JSON payload format is incorrect	Check syntax/format of JSON payload sent in the request.
Error Code: E104 HTTP Response Code: 400 Category: REQVLD Error Description: Internal conversion of content JSON failed with a JSONException.	Internal JSON format data formation failure	Check for any missing parameters in the JSON payload.
Error Code: E105 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid format of data	This is an invalid payload. The parameter with number type is not set correctly.	Check the number/integer parameters for correctness in the payload.
Error Code: E106 HTTP Response Code: 400 Category: REQVLD Error Description: Duplicate subscriber key	The subscriber keys used in CREATE operation are already present on the database.	Check the key which is reported as duplicate. Delete the existing provisioned data if not required and recreate the content. If it is only an UPDATE, use a PUT or PATCH operation.
Error Code: E107 HTTP Response Code: 400 Category: REQVLD Error Description: Schema validation failed and occurrence constraint violation	Subscriber keys in payload crossed the limit. For example: By default five IMSIs can be provisioned for a subscriber. You will encounter this error if a sixth IMSI provisioning is attempted.	Check the limit of keys for a particular key type defined in the schema. Increase the limit or remove the unused keys to provision new keys.
Error Code: E108 HTTP Response Code: 400 Category: REQVLD Error Description: Data insertion failed due to duplicate key for the content	The subscriber keys used in CREATE operation are already present on the database.	Check the key which is reported as duplicated. Delete the existing provisioned data if not required and recreate the content. If it is only an UPDATE, use a PUT or PATCH operation.
Error Code: E109 HTTP Response Code: 400 Category: REQVLD Error Description: Insertion or update of invalid records to database	Internal SQL transaction failures due to invalid records not matching when table schema is inserted/updated.	Check the content shared in payload for insertion/update.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E110 HTTP Response Code: 400 Category: REQULD Error Description: Schema validation failed	Request payload json content does not comply with schema defined on the UDR.	Check the reason shared in the detail with the fixed string Schema Validation failed. ProblemDetails.detail will contain the actual issue due to which the schema validation failed.
Error Code: E111 HTTP Response Code: 400 Category: REQULD Error Description: Invalid schema content supplied	Schema update request payload json content is invalid.	Check if the schema is correct and if there are any formatting issues.
Error Code: E112 HTTP Response Code: 400 Category: REQULD Error Description: Schema version and schema content both should not be empty	Missing mandatory parameters for schema update.	One of the schema version and schema content is empty in the incoming schema update request. This should be fixed.
Error Code: E113 HTTP Response Code: 400 Category: REQULD Error Description: Schema version and schema content both should not be empty	One of schema version and schema content is empty.	One of the schema version and schema content is empty in the incoming schema update request. Either both must be empty or both must not be empty.
Error Code: E114 HTTP Response Code: 400 Category: REQULD Error Description: Subscription already exists	Subscription already present for the subscriber	Update subscription can be performed instead of CREATE or you can DELETE and CREATE the subscription.
Error Code: E115 HTTP Response Code: 400 Category: REQULD Error Description: Invalid monitoredResourceUri	Subscription request is invalid. monitoredResourceUri parameter is incorrectly provided.	Check the correctness and format of monitoredResourceUri parameter.
Error Code: E116 HTTP Response Code: 400 Category: REQULD Error Description: Data is expired	Subscription request expiry is older than the present time.	Update the expiry time to a future timestamp from present time.
Error Code: E117 HTTP Response Code: 400 Category: REQULD Error Description: Schema validation failed, invalid key value	Schema validation failed for incoming request due to invalid subscriber keys.	Check the key values as part of uelds present in the payload. Check the type and format issues and fix accordingly.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E118 HTTP Response Code: 400 Category: REQVLD Error Description: Schema validation failed, Unknown key, Key provided in the request is invalid.	Schema validation failed for incoming request due to invalid subscriber keys.	Check the key values as part of profile data present in the payload. Use the valid keys for provisioning.
Error Code: E119 to E140 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: Reserved for generic payload related errors	NA	NA
Error Code: E141 to E160 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: Reserved for SLF payload related errors	NA	NA
Error Code: E161 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid content, IMEI and IMEISV deletion not allowed	IMEI or IMEISV cannot be deleted using UPDATE request on a EIR subscriber.	Do not send IMEI or IMEISV keys in payload for deletion using UPDATE request.
Error Code: E162 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid content, equipment status cannot be empty.	Equipment status empty in EIR CREATE subscriber request	The equipment-status field for CREATE subscriber must not be empty.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E163 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid content, equipment status cannot have all its content set to false.	Equipment status sent in request is invalid.	Equipment-status must have at least one among blacklisted, whitelisted, or greylisted set to true. For example: <pre>"equipment-status" : { "grey": true, "black": true, "white": true } → correct payload</pre> <pre>"equipment-status" : { "grey": false, "black": false, "white": false }</pre>
Error Code: E164 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid content, equipment status missing for EIR profile.	Equipment status not present in EIR CREATE subscriber request	Check and send the equipment-status as part of the EIR profile creation payload.
Error Code: E165 HTTP Response Code: 400 Category: REQVLD Error Description: Profile creation cannot have both IMEI and IMEISV	EIR profile creation payload cannot have both IMEI and IMEISV.	EIR profile creation payload must have either IMEI or IMEISV. Retain the key type that matched with URI and remove the unmatched key.
Error Code: E166 HTTP Response Code: 400 Category: REQVLD Error Description: EIR profile update cannot have IMEI and IMEISV in the payload	EIR profile update payload cannot have both IMEI and IMEISV.	EIR profile creation payload must have either IMEI or IMEISV. Retain the key type that matched with URI and remove the unmatched key.
Error Code: E167 to E180 HTTP Response Code: 400 Category: REQVLD Error Description: Reserved for EIR payload related issues	NA	NA
Error Code: E181 HTTP Response Code: 400 Category: REQVLD Error Description: Duplicate subscription	Duplicate UDM subscription request	Check for existing subscriptions. Recreate subscription by performing delete or you can update the existing subscription.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E182 HTTP Response Code: 400 Category: REQVLD Error Description: Required key [subscriptionId] not found	Required key [subscriptionId] not found in the request	Pass Required key [subscriptionId]
Error Code: E183 HTTP Response Code: 400 Category: REQVLD Error Description: Subscription ID must contain numeric value enclosed with double quotes	Subscription ID is a non-numeric value.	Set subscription ID to a numeric value enclosed in double quotes.
Error Code: E184 to E200 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: Reserved for UDM payload related issues	NA	NA
Error Code: E201 HTTP Response Code: 400 Category: REQVLD Error Description: Multiple matching rows found	Multiple matching rows found during field lookup using VSA APIs	Fetch the complete field instead of individual fields to see the provisioned data.
Error Code: E202 HTTP Response Code: 400 Category: REQVLD Error Description: The only key present in profile cannot be deleted	The only key present in profile is attempted for delete.	Delete entire subscriber if required using DELETE subscriber APIs.
Error Code: E203 HTTP Response Code: 400 Category: REQVLD Error Description: Field cannot be updated	Mentioned field cannot be updated.	Check the reason in detail parameter and fix request accordingly.
Error Code: E204 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid value for quota-type	Attribute mapping update API for quota migration failed with invalid quota-type.	Set quota-type to either quota or dynamic quota.
Error Code: E205 HTTP Response Code: 400 Category: REQVLD Error Description: Invalid value for fiveg-attribute-placeholder	Attribute mapping update API for quota migration failed with invalid fiveg-attribute-placeholder.	Set fiveg-attribute-placeholder to umData or umDataLimits.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E206 HTTP Response Code: 400 Category: REQVL Error Description: Subscriber auto create failed during schema validation	Schema validation failed during auto creation of subscriber.	Check the payload sent in UPDATE request, it must include all the mandatory parameters and must be in proper format as per schema.
Error Code: E207 HTTP Response Code: 400 Category: REQVL Error Description: Quota migration failure	Quota migration failed during data conversion.	Check all attributes sent in payload are correct. Make sure mapping is valid for all parameters.
Error Code: E208 HTTP Response Code: 400 Category: REQVL Error Description: Attribute mapping is not found for the parameters provided	Quota migration failure during data mapping from 4G to 5G	Check all attributes sent in payload are correct. Make sure mapping is valid for all parameters.
Error Code: E209 to E220 HTTP Response Code: Valid HTTP status code Category: REQVL Error Description: Reserved for PCF and VSA payload related issues	NA	NA
Forbidden Cases	-	-
Error Code: E221 to E230 HTTP Response Code: Valid HTTP status code Category: REQVL Error Description: Generic forbidden (403 status code) cases	NA	NA
Error Code: E231 HTTP Response Code: 403 Category: REQVL Error Description: Subscriber linked to the group	Subscriber linked to the group is attempted for deletion.	SLF groups linked to subscriber cannot be deleted. Remove group mapping to subscriber by subscriber update/deletion and proceed to group deletion.
Error Code: E232 to E240 HTTP Response Code: Valid HTTP status code Category: REQVL Error Description: SLF forbidden (403 status code) cases	NA	NA

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E241 to E250 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: EIR forbidden (403 status code) cases	NA	NA
Error Code: E251 HTTP Response Code: 403 Category: REQVLD Error Description: Modify is not allowed	Modification not allowed for this UDM request	Patch request attempting to update field specified in request is not allowed. Try different operations or avoid this operation.
Error Code: E252 to E260 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: UDM forbidden cases	NA	NA
Error Code: E261 to E270 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: PCF or VSA forbidden cases	NA	NA
Subscriber or Content Not Found Errors	-	-
Error Code: E271 HTTP Response Code: 404 Category: SIG/PROV Error Description: Resource fetch from the database failed.	Exception occurred during resource fetch from database.	Check logs for more information on the failure.
Error Code: E272 HTTP Response Code: 404 Category: SIG/PROV Error Description: Subscriber does not exist	Request sent for a non provisioned subscriber.	Create the subscriber before any operation on it.
Error Code: E273 HTTP Response Code: 404 Category: SIG/PROV Error Description: Data field does not exist.	Request sent for unavailable NF data for a provisioned subscriber.	Provision the NF data before retrieval operations.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E274 HTTP Response Code: 404 Category: SIG/PROV Error Description: Requested data not found.	Request sent for unavailable NF or subscription data for a provisioned subscriber.	Provision the NF data or subscription data before retrieval operations.
Error Code: E275 HTTP Response Code: 404 Category: SIG/PROV Error Description: Subscription does not exist.	Request sent for unavailable subscription info retrieval	Create subscription data before retrieval operations.
Error Code: E276 to E300 HTTP Response Code: Valid HTTP status code Category: SIG/PROV Error Description: Generic Not found cases	NA	NA
Error Code: E301 HTTP Response Code: 404 Category: SIG/PROV Error Description: Group not found.	The request SLF group is not found. SLF group used in subscriber creation not found.	Provision the SLF group before retrieval operations or subscriber creation operations are performed.
Error Code: E302 HTTP Response Code: 404 Category: SIG/PROV Error Description: Group identifier not found.	Group ID not found for the SLF subscriber associated group.	Configure NFGROUPID required for the SLF group associated with the subscriber that is being searched.
Error Code: E303 HTTP Response Code: Category: SIG Error Description: Default group lookup failed. Default group not found.	Default group is enabled but group not available.	Configure a default group (with defaultGroupName) using REST API for SLF Group provisioning.
Error Code: E304 HTTP Response Code: 404 Category: SIG Error Description: Default group lookup failed. Group identifier not found.	Requested group ID not found for the default SLF group.	Update default group with required NFGROUP IDs.
Error Code: E305 to E310 HTTP Response Code: Valid HTTP status code Category: SIG/PROV Error Description: Reserved for SLF specific not found cases.	NA	NA

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E311 HTTP Response Code: 404 Category: SIG Error Description: Provided SUPI or GPSI does not match with PEI association.	EIR Subscriber found but SUPI or GPSI match failed.	Update the SUPI or GPSI to the EIR profile using update operations.
Error Code: E312 HTTP Response Code: 404 Category: SIG Error Description: Equipment not found	EIR subscriber not found.	Perform subscriber provisioning as required.
Error Code: E313 to E320 HTTP Response Code: Valid HTTP status code Category: SIG/PROV Error Description: Reserved for EIR specific not found cases.	NA	NA
Error Code: E331 HTTP Response Code: 404 Category: SIG/PROV Error Description: Data row does not exist	VSA operation failed, requested data not found.	Do a complete get and recheck the existing data.
Error Code: E332 HTTP Response Code: 404 Category: PROV Error Description: Attribute mapping entry not found.	Attribute mapping not found for the requested 4G attribute type.	Use MGM API to creating the required 4G to 5G attribute mapping.
Error Code: E333 to E340 HTTP Response Code: 404 Category: PROV Error Description: Reserved for PCF or VSA specific not found cases	NA	NA
Operation Not Allowed Cases	-	-
Error Code: E341 HTTP Response Code: 405 Category: REQULD Error Description: Operation not allowed	REST API used does not support the requested operation type.	Check <i>Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide</i> for the allowed operations for provided REST API in request.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E342 HTTP Response Code: 405 Category: REQVLD Error Description: Max number of subscriptions reached.	Maximum number of subscriptions reached for a subscriber.	Update maxNumberOfSubscription configuration for creating more subscriptions or delete the unused subscriptions to create new ones.
Error Code: E343 to E370 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: Reserved for generic cases	NA	NA
Error Code: E370 to E400 HTTP Response Code: Valid HTTP status code Category: REQVLD Error Description: Reserved for other categories, if any.	NA	NA
Internal server errors cannot be retried	-	-
Error Code: E401 HTTP Response Code: 500 Category: DB Error Description: Resource accessed in an invalid fashion.	Database fetch failure.	Check logs for more details.
Error Code: E402 HTTP Response Code: 500 Category: DB Error Description: Permission denied to requested resource on the database.	Database access failed due to permission issues.	Check the logs for more details.
Error Code: E403 HTTP Response Code: 500 Category: DB Error Description: SQL Command Execution Error	SQL command failed during execution.	Check the logs for more details.
Error Code: E404 HTTP Response Code: 500 Category: INTRNL Error Description: Internal server error.	Unidentified internal server errors.	Check the logs for more details.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E405 HTTP Response Code: 500 Category: INTRNL Error Description: Failed to get subscriber from source	Failed to get subscriber from 4G.	Check connection issues to source 4G UDR. Make sure that the subscriber is present on source 4G UDR.
Error Code: E406 HTTP Response Code: 500 Category: INTRNL Error Description: Invalid host or port number	Failed to get subscriber from 4G due to invalid host details.	Check host and port details and reconfigure for successful connection.
Error Code: E407 to E430 HTTP Response Code: Valid HTTP status code Category: INTRNL Error Description: Reserved for generic errors, if any	NA	NA
Errors Not Implemented: For SQL syntax issues, ideal cases must not occur	-	-
Error Code: E431 HTTP Response Code: 501 Category: DB Error Description: SQL Command Grammar Error	SQL statement syntax issues	Check logs for more details.
Error Code: E432 to E450 HTTP Response Code: Valid HTTP status code Category: INTRNL Error Description: Reserved for generic errors, if any	NA	NA
Service Unavailable Errors can be Retried	-	-
Error Code: E451 HTTP Response Code: 503 Category: DB Error Description: Connection to the database failed.	Fetching database connection failed internally.	Check the database availability and connectivity.
Error Code: E452 HTTP Response Code: 503 Category: DB Error Description: Database table resource not accessible.	Data retrieval from database failed.	Check the database availability and connectivity. Check the logs for more information on SQL state.

Table B-1 (Cont.) Error Code Catalogue

Error Details	Possible Causes	Corrective Actions
Error Code: E453 HTTP Response Code: 503 Category: DB Error Description: Transient database access issue.	Data retrieval from database failed.	Check the database availability and connectivity. Check the logs for more information on SQL state.
Error Code: E454 HTTP Response Code: 503 Category: DB Error Description: Cannot acquire lock on database resource.	Data retrieval from database failed.	Check the database availability and connectivity. Check the logs for more information on SQL state.
Error Code: E455 HTTP Response Code: 503 Category: DB Error Description: UDR database replication down, SLF group provisioning unavailable.	Database replication failure	Fix database replication issues.
Error Code: E456 to E500 Category: DB HTTP Response Code: Valid HTTP status code Error Description: Reserved for other service unavailable errors, if any	NA	NA

C

Log Attribute Details

The following table lists the log attribute details for UDR. These details are applicable to Provisioning Gateway.

Table C-1 Log Attribute Details

Log Attribute	Details	Example Value	Data Type	Notes (source)
thread	Thread Name Internal by Spring boot	XNIO-1 task-1	String	log4j
level	Log Level of log printed	WARN	String	log4j
loggerName	Class which printed the log	ocudr.udr.services. service.DbHandler	String	log4j
message	Outputs the application supplied message	Subscriber does not exist	String	application
endOfBatch	Log4j2 Internal	false	boolean	log4j
loggerFqcn	Log4j2 Internal	org.apache.logging .slf4j.Log4jLogger	String	log4j
instant	Epoch time	{"epochSecond":15 99703750,"nanoOf Second":21006400 0}	Object	log4j
threadId	Outputs the ID of the thread that generated the logging event, set internally by Log4j2	23	Integer	log4j
threadPriority	Thread Priority set internally by Log4j2	5	Integer	log4j
messageTimestamp	Timestamp when log was printed	21-02-17 07:36:06.343+0000	String	application
application	NF Application Name	ocudr	String	application
engVersion	Engineering version of software	1.10.20	String	application
mktgVersion	Marketing version of software	1.10.20.0.0	String	application
microservice	Microservice name	ocudr-nudr- drservice	String	application
vendor	Vendor Name	Oracle	String	application
subscriberId	SubscriberId for which request received	msisdn-11111111 3	String	application
resourceId	Request Uri	nudr-group-id- map/v1/nf-group- ids	String	application

Table C-1 (Cont.) Log Attribute Details

Log Attribute	Details	Example Value	Data Type	Notes (source)
resultCode	Response statusCode	404	String	application
ocLogId	Inter NF logId for tracing	1613547369374_225_ocudr-ingressgateway-6f585c76d4-tp622	String	application
sbiCorrelationHeader	SBI Correlation Header for request received	msisdn-1111111113	String	application
requestType	request type received	GET	String	application
kubernetes.container_name	Container name generating log	nudr-dr-service	String	fluentd
kubernetes.namespace_name	Namespace of service	ocudr	String	fluentd
kubernetes.pod_name	Pod name	ocudr-nudr-dr-service-7f8c47f5c9-flkmz	String	fluentd
kubernetes.container_image	Container image	cne-170-ga-bastion-1:5000/ocudr/nudr_datarepository_service:1.9.50	String	fluentd
kubernetes.container_image_id	Container image ID	cne-170-ga-bastion-1:5000/ocudr/nudr_datarepository_service@sha256:1141f245a3a437f1423496aebf616a6d3315e22ad09904a868bf1b471759b616	String	fluentd
kubernetes.pod_id	POD id	6dfa91f8-2d0a-4d8c-a339-381ce98264df	String	fluentd
kubernetes.host	Worker node name	cne-170-ga-k8s-node-8	String	fluentd
kubernetes.namespace_id	Unique namespace ID assigned by K8	d932a8ae-54e9-4df1-8e30-e0335f0b1303	String	fluentd

Table C-1 (Cont.) Log Attribute Details

Log Attribute	Details	Example Value	Data Type	Notes (source)
labels	All the labels on pod that generate the logs e.g.	"labels": { "pod-template-hash": "7f8c47f5c9", "app_kubernetes_io/instance": "ocudr", "app_kubernetes_io/managed-by": "Tiller", "app_kubernetes_io/name": "nudur-service", "app_kubernetes_io/part-of": "ocudr", "app_kubernetes_io/version": "1.6.0.0", "helm_sh/chart": "nudur-service-1.9.50", "io_kompose_service": "nudur-service" }	object	fluentd
originHost	diameter client fqdn	diamcli1.oracle.com	String	Application Note: only in diameterproxy and diameter-gateway
originRealm	diameter client realm	oracle.com	String	Application Note: only in diameter-gateway
serviceIndications	diameter service indications for GET operations	"serviceIndications": ["CamiantUserData", "CamiantStateData"]	Array	Application Note: only in diameterproxy

Example: Log Under Analysis

```
{
  "instant":
  {
    "epochSecond": 1613547366,
    "nanoOfSecond": 343417698,
    "thread": "XNIO-1 task-1",
    "level": "WARN",
    "loggerName": "ocudr.udr.services.service.DbHandler",
    "message": "Subscriber does not exist",
    "endOfBatch": false,
    "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger",
    "threadId": 46,
    "threadPriority": 5,
    "messageTimestamp": "21-02-17 07:36:06.343+0000",
    "application": "ocudr",
    "sbiCorrelationHeader": "msisdn-11111113",
    "engVersion": "1.10.20",
    "mktgVersion": "1.10.20.0.0",
    "microservice": "ocudr-nudur-"
  }
}
```

```
drservice", "vendor": "Oracle", "subscriberId": "msisdn-1111111113", "resourceId": "
nudur-group-id-map/v1/nf-group-
ids", "resultCode": "404", "ocLogId": "1613547369374_225_ocudr-
ingressgateway-6f585c76d4-tp622", "requestType": "GET"}
```

Example: Log From Kibana

```
{
  "_index": "logstash-2021-02-19",
  "_type": "_doc",
  "_id": "yyQXuHcBwFjE8wmhickN",
  "_version": 1,
  "_score": 0,
  "_source": {
    "stream": "stdout",
    "docker": {
      "container_id":
        "b1b78faa1043132f77148f16777e60b1db8ae30e9e6b5c2f5af45248063f7d6a"
    },
    "kubernetes": {
      "container_name": "nudur-drservice",
      "namespace_name": "bharathudr1",
      "pod_name": "bharathudr1-nudur-drservice-7bd66864c6-fw7cq",
      "container_image": "cne-172-bastion-1:5000/ocudr/
nudur_datarepository_service:ocLogIdTest1",
      "container_image_id": "cne-172-bastion-1:5000/ocudr/
nudur_datarepository_service@sha256:24875dad7fd363bcb8ec300b007491b2259796e1524
46fd3d707b17f62fcd6b4",
      "pod_id": "ea92c753-48af-4b31-9356-1ce24f27025e",
      "host": "cne-172-k8s-node-10",
      "labels": {
        "pod-template-hash": "7bd66864c6",
        "app_kubernetes_io/instance": "bharathudr1",
        "app_kubernetes_io/managed-by": "Helm",
        "app_kubernetes_io/name": "nudur-drservice",
        "app_kubernetes_io/part-of": "ocudr",
        "app_kubernetes_io/version": "1.6.0.0.0",
        "helm_sh/chart": "nudur-drservice-1.10.20",
        "io_kompose_service": "nudur-drservice"
      },
      "master_url": "https://10.233.0.1:443/api",
      "namespace_id": "260e3d4a-d455-4afc-9cff-9aaf8eba79d1"
    },
    "instant": {
      "epochSecond": 1613701232,
      "nanoOfSecond": 431593919
    },
    "thread": "XNIO-1 task-1",
    "level": "WARN",
    "loggerName": "ocudr.udr.services.service.DbHandler",
    "message": "Subscriber does not exist",
    "endOfBatch": false,
    "loggerFqcn": "org.apache.logging.slf4j.Log4jLogger",
    "threadId": 46,
    "threadPriority": 5,
  }
}
```



```

    "messageTimestamp": "21-02-19 02:20:32.431+0000",
    "application": "bharathudr1",
    "engVersion": "1.10.20",
    "mktgVersion": "1.6.0.0.0",
    "microservice": "bharathudr1-nudr-drservice",
    "vendor": "Oracle",
    "subscriberId": "imsi-1000000002",
    "sbiCorrelationHeader": "imsi-1000000002",
    "resourceId": "nudr-group-id-map-prov/v1/slf-group",
    "resultCode": "404",
    "ocLogId": "1613701211520_2220_bharathudr1-
    ingressgateway-7d7659c58b-5k6b2",
    "requestType": "GET",
    "@timestamp": "2021-02-19T02:20:32.432795705+00:00",
    "tag": "kubernetes.var.log.containers.bharathudr1-nudr-
    drservice-7bd66864c6-fw7cq_bharathudr1_nudr-drservice-
    b1b78faa1043132f77148f16777e60b1db8ae30e9e6b5c2f5af45248063f7d6a.log"
  },
  "fields": {
    "@timestamp": [
      "2021-02-19T02:20:32.432Z"
    ],
    "timestamp": []
  },
  "highlight": {
    "resultCode": [
      "@kibana-highlighted-field@404@/kibana-highlighted-field@"
    ],
    "message": [
      "@kibana-highlighted-field@Subscriber@/kibana-highlighted-field@
      @kibana-highlighted-field@does@/kibana-highlighted-field@ @kibana-highlighted-
      field@not@/kibana-highlighted-field@ @kibana-highlighted-field@exist@/kibana-
      highlighted-field@"
    ],
    "kubernetes.namespace_name": [
      "@kibana-highlighted-field@bharathudr1@/kibana-highlighted-field@"
    ]
  }
}

```

D

List of Error Codes for Diameter SH Interface

The following table provides error code details for Diameter SH Interface:

Table D-1 Error Codes for Diameter SH Interface

Diameter SH Interface	Error Details
PUA	Attribute Value Pair: Result-Code Possible Values: - DIAMETER_SUCCESS (2001) - DIAMETER_UNABLE_TO_COMPLY (5012)
PUA	Attribute Value Pair: Experimental-Result Possible Values: - DIAMETER_ERROR_USER_UNKNOWN (5001) - DIAMETER_ERROR_OPERATION_NOT_ALLOWED (5101) - DIAMETER_ERROR_TRANSPARENT_DATA_OUT_OF_SYNC (5105) - DIAMETER_ERROR_SUBS_DATA_ABSENT(5106) - DIAMETER_MISSING_AVP(5005) - DIAMETER_INVALID_AVP_VALUE(5004) - DIAMETER_AVP_OCCURS_TOO_MANY_TIMES(5009) - DIAMETER_UNABLE_TO_DELIVER(3002)
UDA	Attribute Value Pair: Result-Code Possible Values: - DIAMETER_SUCCESS (2001) - DIAMETER_UNABLE_TO_COMPLY (5012)
UDA	Attribute Value Pair: Experimental-Result Possible Values: - DIAMETER_ERROR_USER_UNKNOWN (5001) - DIAMETER_ERROR_OPERATION_NOT_ALLOWED (5101) - DIAMETER_ERROR_SUBS_DATA_ABSENT(5106) - DIAMETER_MISSING_AVP(5005) - DIAMETER_INVALID_AVP_VALUE(5004) - DIAMETER_AVP_OCCURS_TOO_MANY_TIMES(5009) - DIAMETER_UNABLE_TO_DELIVER(3002)
SNA	Attribute Value Pair: Result-Code Possible Values: - DIAMETER_SUCCESS (2001) - DIAMETER_UNABLE_TO_COMPLY (5012)

Table D-1 (Cont.) Error Codes for Diameter SH Interface

Diameter SH Interface	Error Details
SNA	Attribute Value Pair: Experimental-Result Possible Values: • DIAMETER_ERROR_USER_UNKNOWN (5001) • DIAMETER_ERROR_OPERATION_NOT_ALLOWED (5101) • DIAMETER_ERROR_SUBS_DATA_ABSENT(5106) • DIAMETER_MISSING_AVP(5005) • DIAMETER_INVALID_AVP_VALUE(5004) • DIAMETER_AVP_OCCURS_TOO_MANY_TIMES(5009) • DIAMETER_UNABLE_TO_DELIVER(3002)
CEA	Attribute Value Pair: Result-Code Possible Values: • DIAMETER_UNKNOWN_PEER(3010) • DIAMETER_NO_COMMON_APPLICATION(5010)