Oracle® Communications

Cloud Native Core, Unified Data Repository Troubleshooting Guide

Release 25.2.100
G41785-02
November 2025

Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide, Release 25.2.100

G41785-02

Contents

1 Introduction

1.1 References 1

2 Logs

2.1 Log Levels 1
2.2 Log Attribute Details 1
2.3 Collecting Logs 6
2.4 Configuring Log Levels 6

3 Using Debug Tool

3.1 Enabling Debug Tool 14
3.2 Debug Tool Configuration Parameters 15
3.3 Accessing Debug Tool 18

4 Troubleshooting Unified Data Repository

4.1 Generic Checklist 1
4.2 Database Related Issues 5
4.3 Deployment Related Issues 8
4.3.1 Debugging Pre-Installation Related Issues 8
4.3.2 Debugging Installation Related Issues 8
4.3.2.1 Debugging Pod Creation Failure 8
4.3.2.2 Debugging Pod Startup Failure 15
4.3.2.3 Debugging UDR with Service Mesh Failure 16
4.3.2.4 Debugging SLF Default Group ID related Issues 18
4.3.2.5 Debugging Subscriber Activity Logging 18
4.3.2.6 Debugging Subscriber Bulk Import Tool Related Issues 19
4.3.2.7 Debugging NF Scoring for a Site 20
4.3.2.8 Debugging Subscriber Export Tool Related Issues 20
4.3.2.9 Debugging Controlled Shutdown Related Issues 21
4.3.2.10 Debug Readiness Failure 21
4.3.2.11 Enable cnDBTier Metrics with OSO Prometheus 22

4.3.2.12 Debugging ndbmysqld Pods Restart During cnDBTier Installation or Upgrade 34
4.3.2.13 Debugging Error Logging Related Issues 35
4.3.2.14 Debugging Suppress Notification Related Issues 35
4.3.2.15 Debugging Diameter S13 Interface Related Issues 35
4.3.2.16 Debugging TLS Related Issues 37
4.3.2.17 Debugging Dual Stack Related Issues 38
4.3.2.18 Debugging Lifecycle Management (LCM) Based Automation Issues 39
4.3.2.19 Troubleshooting TLS 1.3 Support for Kubernetes API Server 39
4.3.3 Debugging Post Installation Related Issues 40
4.3.3.1 Debugging Helm Test Issues 40
4.3.3.2 Debugging Horizontal Pod Autoscaler Issues 42
4.3.3.3 Debugging HTTPS Support Related Issues 42
4.3.3.4 Debugging PodDisruptionBudget Related Issues 45
4.3.3.5 Debugging Pod Eviction Issues 46
4.3.3.6 Debugging Taints or Tolerations Misconfigurations 47
4.3.3.7 Debugging UDR Registration with NRF Failure 48
4.3.3.8 Debugging User Agent Header Related Issues 48
4.3.3.9 Debugging LCI and OCI Header Related Issues 49
4.3.3.10 Debugging Conflict Resolution Feature 49
4.3.3.11 Debugging UDR Error Responses Using App Error Code 50
4.3.3.12 Debugging Provisioning Logs Related Issues 51
4.3.4 Debugging Upgrade or Rollback Failure 52
4.4 Service Related Issues 52
4.4.1 Resolving Microservices related Issues through Metrics and ConfigDB 52
4.4.2 Debugging Errors from Egress Gateway 58
4.4.3 Debugging Errors from Ingress Gateway 64
4.4.4 Debugging Errors from nudr-config 64
4.4.5 Debugging Notification Issues 65

5 Alert Configuration

5.1 Alert Details 1
5.1.1 System Level Alerts 4
5.1.1.1 OcudrSubscriberNotFoundAbove1Percent 4
5.1.1.2 OcudrSubscriberNotFoundAbove10Percent 5
5.1.1.3 OcudrSubscriberNotFoundAbove25Percent 5
5.1.1.4 OcudrSubscriberNotFoundAbove50Percent 6
5.1.1.5 OcudrPodsRestart 6
5.1.1.6 NudrServiceDown 7
5.1.1.7 NudrProvServiceDown 8
5.1.1.8 NudrNotifyServiceServiceDown 9

5.1.1.9 NudrNRFClientServiceDown 10
5.1.1.10 NudrConfigServiceDown 11
5.1.1.11 NudrDiameterProxyServiceDown 12
5.1.1.12 NudrOnDemandMigrationServiceDown 13
5.1.1.13 OcudrIngressGatewayServiceDown 14
5.1.1.14 OcudrEgressGatewayServiceDown 15
5.1.1.15 OcudrDbServiceDown 16
5.1.1.16 OcudrIngressGatewayProvServiceDown 16
5.1.2 Application Level Alerts 17
5.1.2.1 OcudrTrafficRateAboveMajorThreshold 17
5.1.2.2 OcudrTrafficRateAboveMinorThreshold 18
5.1.2.3 OcudrTrafficRateAboveCriticalThreshold 19
5.1.2.4 OcudrTransactionErrorRateAbove0.1Percent 19
5.1.2.5 OcudrTransactionErrorRateAbove1Percent 20
5.1.2.6 OcudrTransactionErrorRateAbove10Percent 20
5.1.2.7 OcudrTrafficRateAboveCriticalThreshold 21
5.1.2.8 OcudrTrafficRateAboveMajorThreshold 22
5.1.2.9 OcudrTrafficRateAboveMinorThreshold 22
5.1.2.10 OcudrTransactionErrorRateAbove0.1Percent 23
5.1.2.11 OcudrTransactionErrorRateAbove1Percent 24
5.1.2.12 OcudrTransactionErrorRateAbove10Percent 24
5.1.2.13 OcudrTransactionErrorRateAbove25Percent 25
5.1.2.14 OcudrTransactionErrorRateAbove50Percent 25
5.1.2.15 OcudrXFCCValidationFailureAbove10Percent 26
5.1.2.16 OcudrXFCCValidationFailureAbove20Percent 26
5.1.2.17 OcudrXFCCValidationFailureAbove50Percent 27
5.1.2.18 OcudrOverload60Percent 27
5.1.2.19 OcudrOverload75Percent 28
5.1.2.20 OcudrOverload80Percent 28
5.1.2.21 OcudrOverload90Percent 29
5.1.2.22 SLFSucessTxnDefaultGroupIdRateAbove1Percent 29
5.1.2.23 SLFSucessTxnDefaultGroupIdRateAbove10Percent 29
5.1.2.24 SLFSucessTxnDefaultGroupIdRateAbove25Percent 30
5.1.2.25 SLFSucessTxnDefaultGroupIdRateAbove50Percent 30
5.1.2.26 OcudrDiameterCongestionCongestedState 31
5.1.2.27 OcudrDiameterCongestionDocState 31
5.1.2.28 DRProvServiceOverload60Percent 32
5.1.2.29 DRProvServiceOverload75Percent 32
5.1.2.30 DRProvServiceOverload80Percent 33
5.1.2.31 DRProvServiceOverload90Percent 33
5.1.2.32 Diameter-Gateway pod congestion Danger of congestion state 34
5.1.2.33 Diameter-Gateway pod CONGESTED state 34

5.1.2.34 OcudrProvisioningTrafficRateAboveMajorThreshold 34
5.1.2.35 OcudrProvisioningTrafficRateAboveCriticalThreshold 35
5.1.2.36 OcudrProvisioningTransactionErrorRateAbove25Percent 36
5.1.2.37 OcudrProvisioningTransactionErrorRateAbove50Percent 37
5.1.2.38 PVCFullForSLFExport 37
5.1.2.39 FailedExtractForSLFExport 37
5.1.2.40 BulkImportTransferInFailed 38
5.1.2.41 ExportToolTransferOutFailed 38
5.1.2.42 BulkImportTransferOutFailed 39
5.1.2.43 PVCFullForXMLBulkImport 39
5.1.2.44 PVCFullForBulkImport 39
5.1.2.45 OperationalStatusCompleteShutdown 40
5.1.2.46 NFScoreCalculationFailed 40
5.1.2.47 PVCFullForEXMLExport 41
5.1.2.48 EXMLExportFailed 41
5.1.2.49 IngressgatewayPodProtectionDocState 41
5.1.2.50 IngressgatewayPodProtectionCongestedState 42
5.1.2.51 RetryNotificationRecordsMaxLimitExceeded 42
5.1.2.52 UserAgentHeaderNotFoundMorethan10PercentRequest 43
5.1.2.53 EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold 43
5.1.2.54 EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold 44
5.1.2.55 EgressGatewayJVMBufferMemoryUsedAboveCriticalThreshold 44
5.1.2.56 NudrDiameterGatewayDown 45
5.1.2.57 DiameterPeerConnectionsDropped 45
5.1.2.58 IGWSignallingPodProtectionDOCState 46
5.1.2.59 IGWSignallingPodProtectionCongestedState 47
5.1.2.60 IGWSignallingPodProtectionByRateLimitRejectedRequest 47

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.
italic	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 lists the documentation updates for release 25.2.1xx.

Release 25.2.100 - G41785-02 - November 2025

- Updated the common service release number in the entire document.

Release 25.2.100 - G41785-01 - October 2025

- Updated the release number to 25.2.100 in the entire document.
- Added the TLS 1.3 Support for Kubernetes API Server section.

1 Introduction

This document provides information about troubleshooting Oracle Communications Cloud Native Core Unified Data Repository (UDR) services and managed objects.

> **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.

Overview

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. It provides:

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

This guide provides information about resolving problems that you may experience while installing and configuring UDR. It also provides information about tools that you can use to collect and analyze diagnostic data.

This guide also describes common problems that may arise while installing, configuring, and using UDR. After identifying the issue, perform the required steps to resolve the issue.

1.1 References

Refer the following documents for more information about UDR and its related network functions:

- *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository REST Specification Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository Network Impact Report*

2
Logs

This chapter explains the process to retrieve the logs and status that can be used for effective troubleshooting.

2.1 Log Levels

Logs register system events along with their date and time of occurrence. They also provide important details about a chain of events that could have led to an error or problem.

A log level helps in defining the severity level of a log message. For UDR, the log level of a microservice can be set to any one of the following valid values:

- **TRACE:** A log level that describes events, as a step by step execution of code. This can be ignored during the standard operation, but may be useful during extended debugging sessions.
- **DEBUG:** A log level used for events during software debugging when more granular information is needed.
- **INFO:** A standard log level indicating that something has happened. For example an application has entered a certain state.
- **WARN:** A log level indicates that something unexpected has happened in the application, a problem, or a situation that might disturb one of the processes. But this does not mean that the application has failed. The WARN level should be used in situations that are unexpected, but the code can continue to work.
- **ERROR:** A log level that should be used when an application hits an issue preventing one or more functionalities from functioning.

Using this information, the logs can be filtered based on the system requirements. For instance, if you want to filter the critical information about your system from the informational log messages, set a filter to view messages with only WARN log level in Kibana.

2.2 Log Attribute Details

The following table lists the log attribute details for UDR. These details are applicable to Provisioning Gateway.

Table 2-1 Log Attribute Details

Log Attribute	Details	Example Value	Data Type	Source
thread	Thread Name Internal by Spring boot	XNIO-1 task-1	String	log4j
level	Log Level of the log printed	WARN	String	log4j
loggerName	Class which printed the log	ocudr.udr.services.service.DbHandler	String	log4j

Table 2-1 (Cont.) Log Attribute Details

Log Attribute	Details	Example Value	Data Type	Source
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":1599703750,"nanoOfSecond":210064000}	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-1111111113	String	Application
resourceId	Request Uri	nudr-group-id-map/v1/nf-group-ids	String	Application
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

Table 2-1 (Cont.) Log Attribute Details

Log Attribute	Details	Example Value	Data Type	Source
kubernetes.pod_name	Pod name	ocudr-nudr-drservice-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

labels	All the labels on pod that generate the logs	"labels": {
		"pod-template-hash": "7f8c47f5c9",
		"app_kubernetes_io/instance": "ocudr",
		"app_kubernetes_io/managed-by": "Tiller",
		"app_kubernetes_io/name": "nudr-drservice",
		"app_kubernetes_io/part-of": "ocudr",
		"app_kubernetes_io/version": "1.6.0.0.0",
		"helm_sh/chart": "nudr-drservice-1.9.50",
		"io_kompose_service": "nudr-drservice"
		}
			object	fluentd

Table 2-1 (Cont.) Log Attribute Details

Log Attribute	Details	Example Value	Data Type	Source
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","me
ssage":"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-111111
1113","engVersion":"1.10.20","mktgVersion":"1.10.20.0.0","microservice":"ocudr
-nudr-
drservice","vendor":"Oracle","subscriberId":"msisdn-1111111113","resourceId":"
nudr-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": "nudr-drservice",
      "namespace_name": "bharathudr1",
      "pod_name": "bharathudr1-nudr-drservice-7bd66864c6-fw7cq",
      "container_image": "cne-172-bastion-1:5000/ocudr/
nudr_datarepository_service:ocLogIdTest1",
```

```
      "container_image_id": "cne-172-bastion-1:5000/ocudr/
nudr_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": "nudr-drservice",
        "app_kubernetes_io/part-of": "ocudr",
        "app_kubernetes_io/version": "1.6.0.0.0",
        "helm_sh/chart": "nudr-drservice-1.10.20",
        "io_kompose_service": "nudr-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-100000002",
    "sbiCorrelationHeader": "imsi-100000002",
    "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@"
    ]
  }
}
```

2.3 Collecting Logs

Log files are used to register system events, together with their date and time of occurrence. They can be valuable tools for troubleshooting. Not only do logs indicate that specific events occurred, but also provide important clues about a chain of events that led to an error or problem.

This section describes how to collect logs from pods and containers. The steps are as follows:

1. Run the following command to get the pod details:

   ```
   kubectl -n <namespace_name> get pods
   ```

2. Run the following command to collect the logs from the specific pods or containers:

   ```
   kubectl logs <podname> -n <namespace> -c <containername>
   ```

3. Store the log in a file using the following command:

   ```
   kubectl logs <podname> -n <namespace> > <filename>
   ```

4. (Optional) Run the following command for the log stream with file redirection starting with last 100 lines of log:

   ```
   kubectl logs <podname> -n <namespace> -f --tail <number of lines> >
   <filename>
   ```

   For more information on kubectl commands, see Kubernetes website.

   For more information on how to collect the logs, see *Oracle Communication Cloud Native Core, Data Collector User Guide*.

2.4 Configuring Log Levels

To view logging configurations and update logging levels, use the **Logging Level Configuration** page under **Logging Level** in the CNC Console. For more information, see the **Log Level Configuration** section in the *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

Configuring Migration Tool Log Level Using Custom Value File

Open the custom value file used during installation migration tool to configure as follows. For more information see , “*Migration Tool*” sections in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*:

- To configure pre-migration-hook:

```
 # Pre Install Hook configurations. Used for DB Schema Upgrade
preInstall:
  image:
    name: nudr_pre_migration_hook
    tag: ${nudr_migration_tool_tag}
 root:
   logLevel: WARN
 ocudr:
   logLevel: WARN
```

- To configure logging of migration-tool:

```
# Logging level
logging:
  level:
   root: "WARN"
   ocudr: "WARN"
```

> **Note**
>
> The following log level values are supported:
>
> - DEBUG
> - INFO
> - WARN
> - ERROR

3 Using Debug Tool

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

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

Prerequisites

This section explains the prerequisites for using the debug tool.

> **Note**
>
> - For CNE 23.2.0 and later versions, follow Step a of Configuration in CNE.
> - For CNE versions prior to 23.2.0, follow Step b of Configuration in CNE.

1. **Configuration in CNE**
   Perform the following configurations in the Bastion Host. You need admin privileges to perform these configurations.
   a. When UDR is installed on CNE version 23.2.0 or above

      > **Note**
      >
      > - In CNE version 23.2.0 or above, the default CNE 23.2.0 Kyverno policy, disallow-capabilities, do not allow NET_ADMIN and NET_RAW capabilities that are required for debug tool.
      > - To run Debug tool on CNE 23.2.0 and above, the user must modify the existing Kyverno policy, disallow-capabilities, as below.

      #### Adding a Namespace to an Empty Resource

      i. Run the following command to verify if the current disallow-capabilities cluster policy has namespace in it.
      Example:

      ```
      $ kubectl get clusterpolicies disallow-capabilities -oyaml
      ```

Sample output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  -exclude:
      any:
      -resources:{}
```

ii. If there are no namespaces, then patch the policy using the following command to add <namespace> under resources:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/
resources", "value": {"namespaces":["<namespace>"]} }]'
```

Example:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/
resources", "value": {"namespaces":["ocudr"]} }]'
```

Sample output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  -exclude:
      resources:
        namespaces:
        -ocudr
```

iii. If in case it is needed to remove the namespace added in the above step, use the following command:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "replace", "path": "/spec/rules/0/exclude/any/0/
resources", "value": {} }]'
```

Sample output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  -exclude:
```

```
      any:
      -resources:{}
```

Adding a Namespace to an Existing Namespace List

i. Run the following command to verify if the current disallow-capabilities cluster policy has namespaces in it.
Example:

```
$ kubectl get clusterpolicies disallow-capabilities -oyaml
```

Sample output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  -exclude:
      any:
      -resources:
          namespaces:
          -namespace1
          -namespace2
          -namespace3
```

ii. If there are namespaces already added, then patch the policy using the following command to add <namespace> to the existing list:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/resources/
namespaces/-", "value": "<namespace>" }]'
```

Example:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/resources/
namespaces/-", "value": "ocudr" }]'
```

Sample output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  -exclude:
      resources:
        namespaces:
        -namespace1
        -namespace2
```

```
        -namespace3
        -ocudr
```

iii. If in case it is needed to remove the namespace added in the above step, use the following command:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "remove", "path": "/spec/rules/0/exclude/any/0/
resources/namespaces/<index>"}]'
```

Example:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "remove", "path": "/spec/rules/0/exclude/any/0/
resources/namespaces/3"}]'
```

Sample output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  -exclude:
      resources:
        namespaces:
        -namespace1
        -namespace2
        -namespace3
```

> **Note**
>
> While removing the namespace, provide the index value for namespace within the array. The index starts from '0'.

b. When UDR is installed on CNE version prior to 23.2.0
PodSecurityPolicy (PSP) Creation

Create a PSP by running the following command from the bastion host. The parameters **readOnlyRootFileSystem, allowPrivilegeEscalation, allowedCapabilities** are required by debug container.

> **Note**
>
> Other parameters are mandatory for PSP creation and can be customized as per the CNE environment. The default values are recommended.

```
$ kubectl apply -f - <<EOF

apiVersion: policy/v1beta1
```

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

Role Creation

Run the following command to create a role for the PSP:

```
kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: debug-tool-role
  namespace: cncc
rules:
- apiGroups:
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
  resourceNames:
  - debug-tool-psp
EOF
```

RoleBinding Creation

Run the following command to associate the service account for the UDR namespace with the role created for the PSP:

```
$ kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: debug-tool-rolebinding
  namespace: ocudr
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: debug-tool-role
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: system:serviceaccounts
EOF
```

2. **Configuration in NF specific Helm**
   Following updates must be performed in `custom-values.yaml` file.

   a. Log in to the UDR server.

   b. Open the custom-values file:

   ```
   $ vim <custom-values file>
   ```

   c. Under global configuration, add the following:

   ```
   # Allowed Values: DISABLED, ENABLED
   extraContainers: ENABLED
   debugToolContainerMemoryLimit: 4Gi
   extraContainersVolumesTpl: |
    - name: debug-tools-dir
      emptyDir:
        medium: Memory
        sizeLimit: {{ .Values.global.debugToolContainerMemoryLimit |
   quote }}
   extraContainersTpl: |
     - command:
         - /bin/sleep
         - infinity
       image: {{ .Values.global.dockerRegistry }}/ocdebug-tools:23.2.0
       imagePullPolicy: Always
       name: {{ printf "%s-tools-%s"(include "getprefix".) (include
   "getsuffix".) | trunc 63| trimPrefix "-"| trimSuffix "-"  }}
       resources:
         requests:
           ephemeral-storage: "512Mi"
           cpu: "0.5"
           memory: {{ .Values.global.debugToolContainerMemoryLimit |
   quote }}
         limits:
           ephemeral-storage: "512Mi"
           cpu: "0.5"
           memory: {{ .Values.global.debugToolContainerMemoryLimit |
   ```

```
quote }}
    securityContext:
      allowPrivilegeEscalation: true
      capabilities:
        drop:
        - ALL
        add:
        - NET_RAW
        - NET_ADMIN
      runAsUser: 7000
    volumeMounts:
    - mountPath: /tmp/tools
      name: debug-tools-dir
```

Note

- Debug Tool Container comes up with the default user ID - 7000. If the operator wants to override this default value, it can be done using the `runAsUser` field, otherwise, the field can be skipped.
  Default value: uid=7000(debugtool) gid=7000(debugtool) groups=7000(debugtool)
- In case you want to customize the container name, replace the `name` field in the above values.yaml with the following:

  ```
  name: {{ printf "%s-tools-%s" (include "getprefix" .)
  (include "getsuffix" .) | trunc 63 | trimPrefix "-" |
  trimSuffix "-"  }}
  ```

  This will ensure that the container name is prefixed and suffixed with the necessary values.

For more information on how to customize parameters in the custom yaml value files, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade and Fault Recovery Guide*.

d. Under service specific configurations for which debugging is required, add the following:

```
# Allowed Values: DISABLED, ENABLED, USE_GLOBAL_VALUE
extraContainers: USE_GLOBAL_VALUE
```

> **Note**
>
> - At the global level, `extraContainers` flag can be used to enable/disable injecting extra containers globally. This ensures that all the services that use this global value have extra containers enabled/disabled using a single flag.
> - At the service level, `extraContainers` flag determines whether to use the extra container configuration from the global level or enable/disable injecting extra containers for the specific service.

Run the Debug Tool

To run Debug Tool, perform the following steps:

1. Run the following command to retrieve the POD details:

   ```
   $ kubectl get pods -n <k8s namespace>
   ```

   Example:

   ```
   $ kubectl get pods -n ocudr
   ```

2. Run the following command to enter into Debug Tool Container:

   ```
   $ kubectl exec -it <pod name> -c <debug_container name> -n <namespace> bash
   ```

   Example:

   ```
   $ kubectl exec -it ocudr-nfaccesstoken-49fb96494c-k8w9q -c tools -n ocudr bash
   ```

3. Run the debug tools:

   ```
   bash -4.2$ <debug_tools>
   ```

   Example:

   ```
   bash -4.2$ tcpdump
   ```

4. Copy the output files from container to host:

   ```
   $ kubectl cp -c <debug_container name> <pod name>:<file location in
   container> -n <namespace> <destination location>
   ```

   Example:

   ```
   $ kubectl cp -c tools ocudr-nfaccesstoken-49fb96494c-k8w9q:/tmp/tools/
   capture.pcap -n ocudr /tmp/tools/
   ```

Tools Tested in Debug Container

Following is the list of debug tools that are tested.

Table 3-1 tcpdump

Options Tested	Description	Output	Capabilities
-D	Print the list of the network interfaces available on the system and on which *tcpdump* can capture packets.	`tcpdump -D` 1. eth02. 2. nflog (Linux netfilter log (NFLOG) interface) 3. nfqueue (Linux netfilter queue (NFQUEUE) interface) 4. any (Pseudo-device that captures on all interfaces) 5. lo [Loopback]	NET_ADMIN, NET_RAW
-i	Listen on *interface*	`tcpdump -i eth0` tcpdump: verbose output suppressed, use -v or -vv for full protocol decodelistening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes12:10:37.381199 IP cncc-core-ingress-gateway-7ffc49bb7f-2kkhc.46519 > kubernetes.default.svc.cluster.local.https: Flags [P.], seq 1986927241:1986927276, ack 1334332290, win 626, options [nop,nop,TS val 849591834 ecr 849561833], length 3512:10:37.381952 IP cncc-core-ingress-gateway-7ffc49bb7f-2kkhc.45868 > kube-dns.kube-system.svc.cluster.local.domain: 62870+ PTR? 1.0.96.10.in-addr.arpa. (40)	NET_ADMIN, NET_RAW
-w	Write the raw packets to file rather than parsing and printing them out.	`tcpdump -w capture.pcap -i eth0`	NET_ADMIN, NET_RAW
-r	Read packets from *file* (which was created with the **-w** option).	`tcpdump -r capture.pcap` reading from file /tmp/capture.pcap, link-type EN10MB (Ethernet)12:13:07.381019 IP cncc-core-ingress-gateway-7ffc49bb7f-2kkhc.46519 > kubernetes.default.svc.cluster.local.https: Flags [P.], seq 1986927416:1986927451, ack 1334332445, win 626, options [nop,nop,TS val 849741834 ecr 849711834], length 3512:13:07.381194 IP kubernetes.default.svc.cluster.local.https > cncc-core-ingress-gateway-7ffc49bb7f-2kkhc.46519: Flags [P.], seq 1:32, ack 35, win 247, options [nop,nop,TS val 849741834 ecr 849741834], length 3112:13:07.381207 IP cncc-core-ingress-gateway-7ffc49bb7f-2kkhc.46519 > kubernetes.default.svc.cluster.local.https: Flags [.], ack 32, win 626, options [nop,nop,TS val 849741834 ecr 849741834], length 0	NET_ADMIN, NET_RAW

Table 3-2 ip

Options Tested	Description	Output	Capabilities
addr show	Look at protocol addresses.	`ip addr show` 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaultlink/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00inet 127.0.0.1/8 scope host lovalid_lft forever preferred_lft forever2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group defaultlink/ipip 0.0.0.0 brd 0.0.0.04: eth0@if190: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1440 qdisc noqueue state UP group defaultlink/ether aa:5a:27:8d:74:6f brd ff:ff:ff:ff:ff:ff link-netnsid 0inet 192.168.219.112/32 scope global eth0valid_lft forever preferred_lft forever	--
route show	List routes.	`ip route show` default via 169.254.1.1 dev eth0 169.254.1.1 dev eth0 scope link	--
addrlabel list	List address labels	`ip addrlabel list` prefix ::1/128 label 0 prefix ::/96 label 3 prefix ::ffff:0.0.0.0/96 label 4 prefix 2001::/32 label 6 prefix 2001:10::/28 label 7 prefix 3ffe::/16 label 12 prefix 2002::/16 label 2 prefix fec0::/10 label 11 prefix fc00::/7 label 5 prefix ::/0 label 1	--

Table 3-3 netstat

Options Tested	Description	Output	Capabilities
-a	Show both listening and non-listening (for TCP, this means established connections) sockets.	`netstat -a` Active Internet connections (servers and established)Proto Recv-Q Send-Q Local Address Foreign Address Statetcp 0 0 0.0.0.0:tproxy 0.0.0.0:* LISTENtcp 0 0 0.0.0.0:websm 0.0.0.0:* LISTENtcp 0 0 cncc-core-ingress:websm 10-178-254-194.ku:47292 TIME_WAITtcp 0 0 cncc-core-ingress:46519 kubernetes.defaul:https ESTABLISHEDtcp 0 0 cncc-core-ingress:websm 10-178-254-194.ku:47240 TIME_WAITtcp 0 0 cncc-core-ingress:websm 10-178-254-194.ku:47347 TIME_WAITudp 0 0 localhost:59351 localhost:ambit-lm ESTABLISHEDActive UNIX domain sockets (servers and established)Proto RefCnt Flags Type State I-Node Pathunix 2 [ ] STREAM CONNECTED 576064861	--

Table 3-3 (Cont.) netstat

Options Tested	Description	Output	Capabilities
-l	Show only listening sockets.	`netstat -l` Active Internet connections (only servers)Proto Recv-Q Send-Q Local Address Foreign Address Statetcp 0 0 0.0.0.0:tproxy 0.0.0.0:* LISTENtcp 0 0 0.0.0.0:websm 0.0.0.0:* LISTENActive UNIX domain sockets (only servers)Proto RefCnt Flags Type State I-Node Path	--
-s	Display summary statistics for each protocol.	`netstat -s` Ip:4070 total packets received0 forwarded0 incoming packets discarded4070 incoming packets delivered4315 requests sent outIcmp:0 ICMP messages received0 input ICMP message failed.ICMP input histogram:2 ICMP messages sent0 ICMP messages failedICMP output histogram:destination unreachable: 2	--
-i	Display a table of all network interfaces.	`netstat -i` Kernel Interface tableIface MTU RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flgeth0 1440 4131 0 0 0 4355 0 0 0 BMRUlo 65536 0 0 0 0 0 0 0 0 LRU	--

Table 3-4 curl

Options Tested	Description	Output	Capabilities
-o	Write output to <file> instead of stdout.	`curl -o file.txt http://abc.com/file.txt`	--
-x	Use the specified HTTP proxy.	`curl -x proxy.com:8080 -o http://abc.com/file.txt`	--

Table 3-5 ping

Options Tested	Description	Output	Capabilities
<ip>	Run a ping test to see whether the target host is reachable or not.	`ping 10.178.254.194`	NET_ADMIN, NET_RAW
-c	Stop after sending 'c' number of ECHO_REQUEST packets.	`ping -c 5 10.178.254.194`	NET_ADMIN, NET_RAW
-f (with non zero interval)	Flood ping. For every ECHO_REQUEST sent, a period "." is printed, while for every ECHO_REPLY received a backspace is printed.	`ping -f -i 2 10.178.254.194`	NET_ADMIN, NET_RAW

Table 3-6 nmap

Options Tested	Description	Output	Capabilities

<ip>	Scan for Live hosts, Operating systems, packet filters and open ports running on remote hosts.	`nmap 10.178.254.194`
		
		`Starting Nmap 6.40 ( http://nmap.org ) at 2020-09-29 05:54 UTCNmap scan report for`
		
		`10-178-254-194.kubernetes.default.svc.cluster .local (10.178.254.194)Host is up (0.00046s latency).Not shown: 995 closed portsPORT STATE SERVICE22/tcp open ssh179/tcp open bgp6666/tcp open irc6667/tcp open irc30000/tcp open unknownNmap done: 1 IP address (1 host up) scanned in 0.04 seconds`
			--

Table 3-6 (Cont.) nmap

Options Tested	Description	Output	Capabilities

-v	Increase verbosity level	`nmap -v 10.178.254.194`
		
		`Starting Nmap 6.40 ( http://nmap.org ) at 2020-09-29 05:55 UTC`
		`Initiating Ping Scan at 05:55`
		`Scanning 10.178.254.194 [2 ports]`
		`Completed Ping Scan at 05:55, 0.00s elapsed (1 total hosts)`
		`Initiating Parallel DNS resolution of 1 host. at 05:55`
		`Completed Parallel DNS resolution of 1 host. at 05:55, 0.00s elapsed`
		`Initiating Connect Scan at 05:55`
		`Scanning 10-178-254-194.kubernetes.default.svc.cluster.local (10.178.254.194) [1000 ports]`
		`Discovered open port 22/tcp on 10.178.254.194`
		`Discovered open port 30000/tcp on 10.178.254.194`
		`Discovered open port 6667/tcp on 10.178.254.194`
		`Discovered open port 6666/tcp on 10.178.254.194`
		`Discovered open port 179/tcp on 10.178.254.194`
		`Completed Connect Scan at 05:55, 0.02s elapsed (1000 total ports)`
		`Nmap scan report for 10-178-254-194.kubernetes.default.svc.cluster.local (10.178.254.194)`
		`Host is up (0.00039s latency).`
		`Not shown: 995 closed ports`
		`PORT STATE SERVICE`
		`22/tcp open ssh`
		`179/tcp open bgp`
		`6666/tcp open irc`
		`6667/tcp open irc`
		`30000/tcp open unknown`
		
		`Read data files from: /usr/bin/../share/nmap`
		`Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds`
			--

Table 3-6 (Cont.) nmap

Options Tested	Description	Output	Capabilities

-iL	Scan all the listed IP addresses in a file. Sample file	`nmap -iL sample.txt`
		
		`Starting Nmap 6.40 ( http://nmap.org ) at 2020-09-29 05:57 UTC`
		`Nmap scan report for localhost (127.0.0.1)`
		`Host is up (0.00036s latency).`
		`Other addresses for localhost (not scanned): 127.0.0.1`
		`Not shown: 998 closed ports`
		`PORT STATE SERVICE`
		`8081/tcp open blackice-icecap`
		`9090/tcp open zeus-admin`
		
		`Nmap scan report for 10-178-254-194.kubernetes.default.svc.cluster.local (10.178.254.194)`
		`Host is up (0.00040s latency).`
		`Not shown: 995 closed ports`
		`PORT STATE SERVICE`
		`22/tcp open ssh`
		`179/tcp open bgp`
		`6666/tcp open irc`
		`6667/tcp open irc`
		`30000/tcp open unknown`
		
		`Nmap done: 2 IP addresses (2 hosts up) scanned in 0.06 seconds`
			--

Table 3-7 dig

Options Tested	Description	Output	Capabilities
<ip>	It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried.	`dig 10.178.254.194` **Note**: The IP should be reachable from inside the container.	--
-x	Query DNS Reverse Look-up.	`dig -x 10.178.254.194`	--

3.1 Enabling Debug Tool

The default user ID of the debug tool container is '7000'.

```
Default value: uid=7000(debugtool) gid=7000(debugtool) groups=7000(debugtool)
```

You can retain this default value. However, if you want to override this value, then add the following under **securityContext** in **extraContainersTpl**.

```
runAsUser: <user-id>
```

To enable a debug tool container, edit the UDR custom-values.yaml file at the:

- Global Level: Enable the **extraContainers** parameter and configure the **extraContainersTpl** value. A sample code snippet is as follows:

```
extraContainers: ENABLED
extraContainersTpl: |
  - command:
      - /bin/sleep
      - infinity
    image: {{ .Values.global.dockerRegistry }}/debug-tools:1.2.0
    imagePullPolicy: Always
    name: {{ printf "%s-tools-%s" (include "getprefix" .) (include
"getsuffix" .) | trunc 63 | trimPrefix "-" | trimSuffix "-"  }}
    resources:
      limits:
        ephemeral-storage: "4Gi"
        cpu: "1"
        memory: "2Gi"
      requests:
        ephemeral-storage: "2Gi"
        cpu: "0.5"
        memory: "1Gi"
    securityContext:
      allowPrivilegeEscalation: true
      capabilities:
        drop:
        - ALL
        add:
        - NET_RAW
        - NET_ADMIN
      readOnlyRootFilesystem: false
      runAsUser: 1002
```

- Service Level: Set the **extraContainers: USE_GLOBAL_VALUE/ENABLED** parameter under each microservice section.

3.2 Debug Tool Configuration Parameters

Following are the parameters used to configure debug tool.

CNE Parameters

Table 3-8 CNE Parameters

Parameter	Description
apiVersion	APIVersion defines the version schema of this representation of an object.
kind	Kind is a string value representing the REST resource this object represents.
metadata	Standard object's metadata.
metadata.name	Name must be unique within a namespace.

Table 3-8 (Cont.) CNE Parameters

Parameter	Description
spec	spec defines the policy enforced.
spec.readOnlyRootFilesystem	Controls whether the containers run with a read-only root filesystem (that is, no writable layer).
spec.allowPrivilegeEscalation	Gates whether or not a user is allowed to set the security context of a container to allowPrivilegeEscalation=true.
spec.allowedCapabilities	Provides a list of capabilities that are allowed to be added to a container.
spec.fsGroup	Controls the supplemental group applied to some volumes. RunAsAny allows any fsGroup ID to be specified.
spec.runAsUser	Controls which user ID the containers are run with. RunAsAny allows any runAsUser to be specified.
spec.seLinux	RunAsAny allows any seLinuxOptions to be specified.
spec.supplementalGroups	Controls which group IDs containers add. RunAsAny allows any supplementalGroups to be specified.
spec.volumes	Provides a list of allowed volume types. The allowable values correspond to the volume sources that are defined when creating a volume.

Role Creation Parameters

Table 3-9 Role Creation

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

Table 3-10 Role Binding Creation

Parameter	Description
apiVersion	APIVersion defines the versioned schema of this representation of an object.
kind	Kind is a string value representing the REST resource this object represents.

Table 3-10 (Cont.) Role Binding Creation

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

Debug Tool Configuration Parameters

Table 3-11 Debug Tool Configuration Parameters

Parameter	Description
extraContainers	Specifies the spawns debug container along with application container in the pod.
debugToolContainerMemoryLimit	Indicates the memory assigned for the debug tool container.
extraContainersVolumesTpl	Specifies the extra container template for the debug tool volume.
extraContainersVolumesTpl.name	Indicates the name of the volume for debug tool logs storage.
extraContainersVolumesTpl.emptyDir.medium	Indicates the location where `emptyDir` volume is stored.
extraContainersVolumesTpl.emptyDir.sizeLimit	Indicates the `emptyDir` volume size.
command	String array used for container command.
image	Docker image name
imagePullPolicy	Image Pull Policy
name	Name of the container
resources	Compute Resources required by this container
resources.limits	Limits describes the maximum amount of compute resources allowed
resources.requests	Requests describes the minimum amount of compute resources required
resources.limits.cpu	CPU limits
resources.limits.memory	Memory limits
resources.limits.ephemeral-storage	Ephemeral Storage limits
resources.requests.cpu	CPU requests
resources.requests.memory	Memory requests
resources.requests.ephemeral-storage	Ephemeral Storage requests

Table 3-11 (Cont.) Debug Tool Configuration Parameters

Parameter	Description
securityContext	Security options the container should run with.
securityContext.allowPrivilegeEscalation	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This directly controls if the no_new_privs flag will be set on the container process
secuirtyContext.readOnlyRootFilesystem	Whether this container has a read-only root filesystem. Default is false.
securityContext.capabilities	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
securityContext.capabilities.drop	Removed capabilities
secuirtyContext.capabilities.add	Added capabilities
securityContext.runAsUser	The UID to run the entrypoint of the container process.
volumeMounts.mountPath	Indicates the path for volume mount.
volumeMounts.name	Indicates the name of the directory for debug tool logs storage.

3.3 Accessing Debug Tool

After installation, the debug tool gets injected into the pods. A sample pod output is as follows:

Debug Tool Container Injection

```
[root@master ~]# kubectl get pods -n myudr
NAME                                                READY   STATUS
RESTARTS   AGE
ocudr-ingressgateway-7cfd68cbbd-6t5rm               2/2     Running
0          2d6h
ocudr-nudr-diameterproxy-54b7c4996c-rfdvz           2/2     Running
0          3d6h
ocudr-nudr-diameterproxy-54b7c4996c-sx8lt           2/2     Running
0          2d5h
ocudr-nudr-drservice-5b48776fb-k6gzw                2/2     Running
0          111m
ocudr-nudr-ondemand-migration-68957f665c-mqwns      2/2     Running
0          104m
```

To access the debug tool:

1. Run the following command to enter the debug tool container:

   ```
   kubectl exec -it <pod name> -c <container name> -n <namespace> bash
   ```

   Example: `kubectl exec -it ocudr-ingressgateway-7cfd68cbbd-6t5rm -c tools -n myudr bash`

2. Run the following command to copy output files from container to host:

   ```
   kubectl cp -c <container name> <pod name>:<file location in container> -n
   <namespace> <destination location>
   ```

   Example: `kubectl cp -c tools ocudr-ingressgateway-7cfd68cbbd-6t5rm:/tmp/capture.pcap -n myudr /tmp/`

4
Troubleshooting Unified Data Repository

This chapter provides information to troubleshoot the common errors, which can be encountered during the preinstallation, installation, upgrade, and rollback procedures of Oracle Communications Cloud Native Core, Unified Data Repository (UDR).

4.1 Generic Checklist

The following sections provide generic checklist for troubleshooting UDR:

Deployment Related Checklist

- Run the following command to check the installation of kubectl.

  ```
  $ kubectl
  ```

  If kubectl is not installed, you can visit https://kubernetes.io/docs/tasks/tools/install-kubectl/
- Run the following command to check the installation of UDR.

  ```
  $ kubectl get pods -n <ocudr-namespace>
  ```

Figure 4-1 Sample Output: UDR Pods Status

```
[root@master ~]# kubectl get pods -n myudr
NAME                                        READY   STATUS    RESTARTS   AGE
ocudr-alternate-route-864d6d67dc-zxzf8      2/2     Running   0          12d
ocudr-egressgateway-cd9ccb8cc-4kdhx         2/2     Running   0          12d
ocudr-ingressgateway-645c696f4d-26flf       2/2     Running   0          12d
ocudr-nudr-config-8d9df8f7d-cbdm9           2/2     Running   0          12d
ocudr-nudr-config-server-579c68c4d4-w2nmm   2/2     Running   0          12d
ocudr-nudr-diameterproxy-84fcbfb4d7-th6r8   2/2     Running   0          12d
ocudr-nudr-drservice-5c47898d77-5ndjd       2/2     Running   0          12d
ocudr-nudr-notify-service-6b6f95db79-fgzfj  2/2     Running   0          12d
ocudr-nudr-nrf-client-service-5d6fff5c7-hwsds 2/2   Running   0          12d
```

> **Note**
>
> The **STATUS** of all the pods is 'Running'.

- Run the following command to view all the events related to a particular namespace.

  ```
  kubectl get events -n <ocudr-namespace>
  ```

- Ensure the preinstall job is in the completed state and all the UDR microservices are in the running state.
- To verify the database and user creation, the following guidelines must be followed:

- If the preinstall pod is in the 'ERROR' state, run the following command to check the logs to debug the issue.

  ```
  kubectl logs -n <namespace> <pre install pod name>
  ```

- If you see the following message in logs, it is possibly because the MySQL server does not allow remote connections to the privileged users.

  ```
  {"thrown":{"commonElementCount":0,"localizedMessage":"Access denied for
  user 'root'@'10.233.118.132' to database 'saqdb'","message":"Access
  denied for user 'root'@'%' to database
  'saqdb'","name":"java.sql.SQLSyntaxErrorException","extendedStackTrace":
  "java.sql.SQLSyntaxErrorException: Access denied for user 'root'@'%' to
  database 'saqdb'\n\tat
  ```

  To fix the user access error, run the following steps on all the SQL nodes to modify the user table in MySQL DB.

  ```
  1. mysql> update mysql.user set host='%' where User='<privileged
  username>';
     Query OK, 0 rows affected (0.00 sec)
     Rows matched: 1  Changed: 0  Warnings: 0
  2. mysql> flush privileges;
     Query OK, 0 rows affected (0.06 sec)
  ```

- If the preinstall job is complete but the dr-service and notify-service pods are crashing with a similar error message in logs as above, then the user may not be created. To fix this, you need to set the value of **createUser** field in the custom-values.yaml file to 'true' before installing UDR.

  > **Note**
  >
  > For more information on creating a database user, see the **Creating Database User or Group** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

  ```
  preInstall:
      image:
        name: nudr_pre_install_hook
        tag: 25.2.100
      config:
        logLevel: WARN
      # Flag to enable user creation. Keep this flag true.
      # Change to false when installed with vDBTier. For vDBTier
  instllation user creation on DB
      # should be manually done
      createUser: true
  ```

- If the preinstall pod is in the ERROR state with the following error message in logs:

  ```
  "message":"Exception encountered during context initialization -
  cancelling refresh attempt:
  org.springframework.beans.factory.BeanCreationException: Error creating
  bean with name 'createUser': Invocation of init method failed; nested
  exception is java.sql.SQLException: NDB_STORED_USER privilege is not
  supported. Please use MySQL version 8.0.22 or higher",
  ```

  Then, it could be because the data tier you are trying to connect has a MySQL package installed that does not support the NDB_STORED_USER privilege. To fix this, set the **createUser** flag to 'false' and create the user manually on all SQL nodes.

- If there is **"The database secret is empty"** or ***"Invalid data present in the secret"*** error message in the preinstall hook logs, then create the secret as mentioned in the **Installing Unified Data Repository** chapter in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide* . Check for the case sensitivity of the keys in the secret. For example, encryptionKey, dsusername, and so on.

- Run the following command to verify whether UDR specific pods are working as expected:

  ```
  $ kubectl get pods -n <ocudr-namespace>
  ```

Figure 4-2 Sample Output: UDR Pods Status

```
[cloud-user@cne-2230-rc2-bastion-1 ~]$ kubectl get pods -n manisha
NAME                                                  READY   STATUS    RESTARTS        AGE
export-nudr-export-tool-0                             1/1     Running   0               23h
ocudr-alternate-route-5b8fbd4df7-zvf6g                1/1     Running   0               4m58s
ocudr-egressgateway-9c94f54b9-nkngg                   1/1     Running   0               4m59s
ocudr-ingressgateway-prov-76f6cbf4b9-kq7j2            1/1     Running   0               4m58s
ocudr-ingressgateway-sig-866b6b78c9-gl5h5             1/1     Running   0               4m59s
ocudr-nudr-config-6d869dbbfb-26qfz                    1/1     Running   0               4m59s
ocudr-nudr-config-server-65cdcd85bd-czspg             1/1     Running   0               4m59s
ocudr-nudr-diam-gateway-0                             1/1     Running   0               4m59s
ocudr-nudr-diameterproxy-74d96cf84f-tfqdh             1/1     Running   0               4m58s
ocudr-nudr-dr-provservice-646bc659c9-nmpz6            1/1     Running   0               4m59s
ocudr-nudr-drservice-7f447bbccb-7mgrs                 1/1     Running   0               4m59s
ocudr-nudr-notify-service-7f4b749cfb-v4wp5            1/1     Running   0               4m58s
ocudr-nudr-nrf-client-nfmanagement-867c74f77c-2rhwj   1/1     Running   0               4m43s
ocudr-nudr-nrf-client-nfmanagement-867c74f77c-442gc   1/1     Running   1 (3m15s ago)   4m58s
ocudr-performance-5496c69d58-j9kq9                    1/1     Running   0               4m58s
```

Result: In the figure given above, you can see that the status of all the pods is 'Running'.

> **Note**
>
> The number of pods for each service depends on Helm configuration. In addition, all pods must be in a ready state and you need to ensure that there are no continuous restarts.

Helm Installation Checklist

Run the following command to check the installation of helm.

```
$ helm ls
```

If helm is not installed, run the following set of commands one after another to install helm:

1. `curl -o /tmp/helm.tgz https://storage.googleapis.com/kubernetes-helm/helm-v2.9.1-linux-amd64.tar.gz.`

   Replace with the latest Helm download link.

2. `tar -xzvf /tmp/helm.tgz -C /usr/local/bin --strip-components=1 linux-amd64/`
   `helmrm -f /tmp/helm.tgz`

3. `kubectl create serviceaccount --namespace kube-system tiller`

4. `kubectl create clusterrolebinding tiller-cluster-rule --`
   `clusterrole=cluster-admin --serviceaccount=kube-system:tiller`

5. `helm init --service-account tiller`

6. `kubectl get po -n kube-system`

   # Wait for the tiller pod to be up

7. `helm ls`

   # Does not return any error. Try again if an error is returned as the tiller pod may be coming up.

8. `helm install`

   . If this command fails immediately with a syntax error, check for the required data for the helm install command to run.

Database Related Checklist

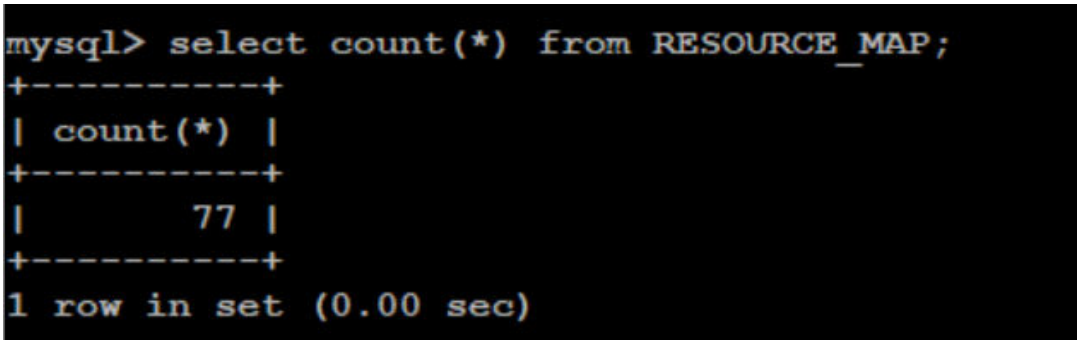
To verify database connectivity:

- Log in to the NDB cluster and verify the creation of UDR database with all the tables. To check the entries in the database tables, run the following command:

  ```
  select count(*) from RESOURCE_MAP
  ```

  It ensures that the connection is fine and the database is created successfully. This count differs based on the **udrServices** option selected under the global section of the custom-values.yaml file. But this table cannot be empty.

Figure 4-3 Sample Output: Verifying Table Entries in Database



- To verify UDR subscribers, check the provisioning flow on UDR. Use the following provisioning URL supported on UDR to verify the provisioning flow:
  - If you use external tools like postman and http2 curl, then follow this URL:
    `http://<ocudr-ingress-gateway-ip>:<http-external-port>/nudr-dr-prov/v1/profile-data/msisdn-1111111113`

    In case of curl, the client must support an http2 curl utility.
  - If HTTPS is enabled in UDR Ingress Gateway, then follow this URL:
    `https://<ocudr-ingress-gateway-ip>:<https-external-port>/nudr-dr-prov/v1/profile-data/msisdn-1111111113`

  Verifying provisioning flow on UDR also confirms the udrdb status on the NDB cluster.
- Check the nudr-nrf-client-nfmanagement logs for no 503 errors. This helps to find out if all the FQDN configured, as part of helm configurations, in values are resolvable.
- Verify NRF registration by checking the **nrfclient_current_nf_status** and **nrfclient_nf_status_with_nrf** metrics on Prometheus.

4.2 Database Related Issues

This section describes the database related issues.

Verifying SQL Exception Failures with nudr-pre-install-hook pod

The **nudr-pre-install-hook** pod creates UDR database along with the tables required. If it does not create the database, then perform the following steps to debug the pod failure.

- Verify whether the **helm install** command hangs for longer time or fails with the BackOffLimit Exceeded error.
- Watch the **kubectl get pods** command based on the release namespace.
- Check whether **nudr-preinstall** pod is going to error state. This means the DB creation has failed or connection to DB is not successful.
- Run the following command on logs:

  ```
  kubectl logs <udr-pre-install-hook pod id> --n <ocudr-namespace>
  ```

- Check the log output of the pods for any warning or SQL exceptions using above command continuously. If any warning or SQL exception is found, it means there is an issue with the SQL connection or the SQL Node. Examine each exception thoroughly to find the root cause.

- Verify the following information in the values.yaml file.

```
global:

  ...

  ...

  ...

  # MYSQL Connectivity Configurations
  mysql:
    dbServiceName: &dbHostName "mysql-connectivity-service.occne-ndb"
#This is a read only parameter. Use the default value.
    port: &dbPortNumber "3306"
    configdbname: &configdbname udrconfigdb
    dbname: &dbname udrdb
    # Do not change the below values
    dbUNameLiteral: &dbUserName dsusername
    dbPwdLiteral: &dbUserPass dspassword
    dbEngine: &dbEngine NDBCLUSTER

  nrfClientDbName: *configdbname
  dbCredSecretName: &dbSecretName 'ocudr-secrets
```

- Ensure that the following service is available in the Cloud Native Environment (CNE).

Figure 4-4 Service Availability in CNE

```
[root@master ocudr]# kubectl get svc -n occne-infra
NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
mysql-connectivity-service  ClusterIP   10.109.123.205   <none>        3306/TCP   3h49m
```

- Check whether Kubernetes secrets are present. If secrets exist, then check their encrypted details like username, password, and DB name. If these details do not exist, then update the secrets.
- After making any changes, run the following command to upgrade Helm.

```
helm upgrade <helm chart> [--version <OCUDR version>] --name <release> --
namespace <ocudr-namespace> -f <ocudr_values.yaml>
```

For more information, see the **Creating Kubernetes Secret - DBName, Username, Password, and Encryption Key** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide* .

Verifying SQL Exception Failures with Common Services pre-install-hook pod

- Run the following command on logs:

```
kubectl logs <failed-pre-install-hook-pod> -n <ocudr-namespace>
```

- Check the log output of the pods for any warning or SQL exceptions using above command continuously. If any warning or SQL exception is found, it means there is an issue with the SQL connection or the SQL Node. Examine each exception thoroughly to find the root cause.

- Verify the following information in the values.yaml file.

```
global:

  ...

  ...

  ...

  # MYSQL Connectivity Configurations
  mysql:
    dbServiceName: &dbHostName "mysql-connectivity-service.occne-ndb"
#This is a read only parameter. Use the default value.
    port: &dbPortNumber "3306"
    configdbname: &configdbname udrconfigdb
    dbname: &dbname udrdb
    # Do not change the below values
    dbUNameLiteral: &dbUserName dsusername
    dbPwdLiteral: &dbUserPass dspassword
    dbEngine: &dbEngine NDBCLUSTER

  nrfClientDbName: *configdbname
  dbCredSecretName: &dbSecretName 'ocudr-secrets'
```

- Ensure that the following service is available in the Cloud Native Environment (CNE).

Figure 4-5 Service Availability in CNE

```
[root@master ocudr]# kubectl get svc -n occne-infra
NAME                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
mysql-connectivity-service   ClusterIP   10.109.123.205   <none>        3306/TCP   3h49m
```

- Check whether Kubernetes secrets are present. If secrets exist, then check their encrypted details like username, password, and DB name. If these details do not exist, then update the secrets.
- After making any changes, run the following command to upgrade Helm.

```
helm install <helm chart> [--version <OCUDR version>] --name <release> --
namespace <ocudr-namespace> -f <ocudr_values.yaml>
```

For more information, see the **Creating Kubernetes Secret - DBName, Username, Password, and Encryption Key** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide* .

Verifying SQL Exception Failure with nudr-pre-upgrade-hook pod

The **nudr-pre-upgrade-hook** pod takes care of the database schema upgrade of UDR. It adds new tables if required, along with few more entries to the existing tables. Perform the following steps to debug this pod failure when there is an issue with the database upgrade:

1. Checks whether the **helm upgrade** command hangs for long time or fails with BackOffLimit exceeded error.
2. Ensure that the **pre_upgrade_hook.yaml** file is present in the templates directory of the target charts, with the required annotation. This is for the nudr-pre-upgrade-hook pod to come up.

```
"helm.sh/hook": "pre-upgrade"
```

3. Watch the **kubectl get pods** command based on the release namespace.
4. Run the following command on the pods to check if the **nudr-pre-upgrade** pod is going to error state. It means that the DB schema upgrade has failed or connection to DB is not successful.
   `kubectl logs <nudr-pre-upgrade-hook pod id> --n <ocudr-namespace>`
5. Check the log output of the pod for any warning or SQL Exception. If there is any, it means there is an issue with the SQL connection or the SQL Node. Check the Exception details to get the root cause.
6. After the upgrade completes, run the following command to verify whether all the pods are running containers with the updated images.
   `kubectl describe pod <pod id> --n <ocudr-namespace>`
7. If the nudr-pre-upgrade pod throws an error, check the logs. If the logs has **"Change in UDR Mode not allowed"** error, then check if the configuration of udrServices in the values.yaml file is different from previous version. If the logs has **"Change in VSA Level not allowed"** error, then check if the configuration of vsaLevel in the values.yaml file is different from previous version.

4.3 Deployment Related Issues

This section describes the most common deployment related issues and their resolution steps. Users are recommended to attempt the resolution provided in this section before contacting Oracle Support.

4.3.1 Debugging Pre-Installation Related Issues

As of now, there are no known preinstallation related issues that you may encounter before installing UDR. However, it is recommended to see the **Prerequisites** and **PreInstallation Tasks** section in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide* to prepare for UDR installation.

4.3.2 Debugging Installation Related Issues

This section describes how to troubleshoot the installation related issues. It is recommended to see the Generic Checklist section also in addition to the information shared in this section.

4.3.2.1 Debugging Pod Creation Failure

A pod creation can fail due to various reasons. Some of the possible scenarios are as follows:

Verifying Pod Image Correctness

To verify pod image:

- Check whether any of the pods is in the **ImagePullBackOff** state.
- Check whether the image name used for any pod is incorrect. Verify the following values in the custom-values.yaml file.

```
global:
  dockerRegistry: ocudr-registry.us.oracle.com:5000/ocudr

nudr-drservice:
```

```
  image:
    name: nudr_datarepository_service
    tag: 25.2.100

nudr-dr-provservice:
  image:
    name: nudr_datarepository_service
    tag: 25.2.100


nudr-notify-service:
  image:
    name: nudr_notify_service
    tag: 25.2.100

nudr-config:
  image:
    name: nudr_config
    tag: 25.2.100

config-server:
  # Image details
  image: ocpm_config_server
  imageTag: 25.2.102
  pullPolicy: IfNotPresent

ingressgateway-sig:
  image:
    name: ocingress_gateway
    tag: 25.2.104
  initContainersImage:
    name: configurationinit
    tag: 25.2.104
  updateContainersImage:
    name: configurationupdate
    tag: 25.2.104
  dbHookImage:
    name: common_config_hook
    tag: 25.2.104
    pullPolicy: IfNotPresent

ingressgateway-prov:
  image:
    name: ocingress_gateway
    tag: 25.2.104
  initContainersImage:
    name: configurationinit
    tag: 25.2.104
  updateContainersImage:
    name: configurationupdate
    tag: 25.2.104
  dbHookImage:
    name: common_config_hook
    tag: 25.2.104
    pullPolicy: IfNotPresent
```

```
egressgateway:
  image:
    name: ocegress_gateway
    tag: 25.2.104
  initContainersImage:
    name: configurationinit
    tag: 25.2.104
  updateContainersImage:
    name: configurationupdate
    tag: 25.2.104
  dbHookImage:
    name: common_config_hook
    tag: 25.2.104
    pullPolicy: IfNotPresent

nudr-diameterproxy
  image:
    name: nudr_diameterproxy
    tag: 25.2.100

nudr-ondemandmigration:
  image:
    name: nudr_ondemandmigration
    tag: 25.2.100

alternate-route:
  deploymentDnsSrv:
    image: alternate_route
    tag: 25.2.104
    pullPolicy: IfNotPresent
  dbHookImage:
    name: common_config_hook
    tag: 25.2.104
    pullPolicy: IfNotPresent

perf-info:
  image: perf-info
  imageTag: 25.2.102
  imagepullPolicy: Always

  dbHookImage:
    name: common_config_hook
    tag: 25.2.104
    pullPolicy: IfNotPresent

app-info:
  image: app-info
  imageTag: 25.2.102
  imagepullPolicy: Always

  dbHookImage:
    name: common_config_hook
    tag: 25.2.104
    pullPolicy: IfNotPresent

nrf-client:
```

```
  image: nrf-client
  imageTag: 25.2.102
  imagepullPolicy: Always

  dbHookImage:
    name: common_config_hook
    tag: 25.2.104
    pullPolicy: IfNotPresent

nudr-dbcr-auditor-service:
  image:
    name: nudr_dbcr_auditor_service
    tag: 25.2.100
    pullPolicy: IfNotPresent
```

- After updating the values.yaml file, run the following command for helm upgrade:
  `helm upgrade <helm chart> [--version <OCUDR version>] --name <release> --namespace <ocudr-namespace> -f <ocudr_values.yaml>`
- If the helm install command is stuck for a long time or fails with timeout error, verify whether the pre install hooks have come up. Verify whether there exists any **ImagePullBackOff** status check as follows.
  hookImageDetails

```
global:
  dockerRegistry: ocudr-registry.us.oracle.com:5000/ocudr

  preInstall:
    image:
      name: nudr_common_hooks
      tag: 25.2.100

  preUpgrade:
    image:
      name: nudr_common_hooks
      tag: 25.2.100

  postUpgrade:
    image:
      name: nudr_common_hooks
      tag: 25.2.100

  postInstall:
    image:
      name: nudr_common_hooks
      tag: 25.2.100

  preRollback:
    image:
      name: nudr_common_hooks
      tag: 25.2.100

  postRollback:
    image:
      name: nudr_common_hooks
      tag: 25.2.100
```

```
  test:
    image:
      name: nf_test
      tag: 25.2.102
```

After updating these values, you can purge the deployment and install helm again.

Verifying Resource Allocation Failure

To verify any resource allocation failure:

- Run the following command to verify whether any pod is in the Pending state.
  `kubectl describe <nudr-drservice pod id> --n <ocudr-namespace>`
- Verify whether any warning on insufficient CPU exists in the describe output of the respective pod. If it exists, it means there are insufficient CPUs for the pods to start. Address this hardware issue.
- If any preinstall hooks are in pending state, then check the resources allocated for hooks. Do not allocate higher values for hooks. If hooks with lower CPU or memory are going to pending state, then there is an issue with available resources on cluster. Check the resources and reduce the number of CPUs alloted to the pod in the values.yaml file.
  hookresources

  ```
  global:
    hookJobResources:
      limits:
        cpu: 2
        memory: 2Gi
      requests:
        cpu: 1
        memory: 1Gi
  ```

  resources

  ```
  nudr-drservice:
    resources:
      limits:
        cpu: 2
        memory: 2Gi
      requests:
        cpu: 2
        memory: 2Gi

  nudr-dr-provservice:
    resources:
      limits:
        cpu: 2
        memory: 2Gi
      requests:
        cpu: 2
        memory: 2Gi

  nudr-notify-service:
    resources:
  ```

```
    limits:
      cpu: 2
      memory: 2Gi
    requests:
      cpu: 2
      memory: 2Gi

nudr-config:
  resources:
    limits:
      cpu: 2
      memory: 2Gi
    requests:
      cpu: 2
      memory: 2Gi

config-server:
  resources:
    limits:
      cpu: 2
      memory: 2Gi
    requests:
      cpu: 2
      memory: 512Mi

nudr-client:
  resources:
    limits:
      cpu: 1
      memory: 2Gi
    requests:
      cpu: 1
      memory: 512Mi

nudr-diameterproxy:
  resources:
    limits:
      cpu: 3
      memory: 4Gi
    requests:
      cpu: 3
      memory: 4Gi

nudr-ondemand-migration:
  resources:
    limits:
      cpu: 2
      memory: 2Gi
    requests:
      cpu: 2
      memory: 2Gi

ingressgateway:
  resources:
    limits:
      cpu: 2
```

```
      memory: 2Gi
      initServiceCpu: 1
      initServiceMemory: 1Gi
      updateServiceCpu: 1
      updateServiceMemory: 1Gi
      commonHooksCpu: 1
      commonHooksMemory: 1Gi
    requests:
      cpu: 2
      memory: 2Gi
      initServiceCpu: 1
      initServiceMemory: 1Gi
      updateServiceCpu: 1
      updateServiceMemory: 1Gi
      commonHooksCpu: 1
      commonHooksMemory: 1Gi

egressgateway:
  resources:
    limits:
      cpu: 2
      memory: 2Gi
      initServiceCpu: 1
      initServiceMemory: 1Gi
      updateServiceCpu: 1
      updateServiceMemory: 1Gi
      commonHooksCpu: 1
      commonHooksMemory: 1Gi
    requests:
      cpu: 2
      memory: 2Gi
      initServiceCpu: 1
      initServiceMemory: 1Gi
      updateServiceCpu: 1
      updateServiceMemory: 1Gi
      commonHooksCpu: 1
      commonHooksMemory: 1Gi

alternate-route:
  resources:
    limits:
      cpu: 2
      memory: 2Gi
      commonHooksCpu: 1
      commonHooksMemory: 1Gi
    requests:
      cpu: 2
      memory: 2Gi
      commonHooksCpu: 1
      commonHooksMemory: 1Gi

perf-info:
  resources:
    limits:
      cpu: 1
      memory: 1Gi
```

```
    requests:
      cpu: 1
      memory: 1Gi

app-info:
  resources:
    limits:
      cpu: 1
      memory: 1Gi
    requests:
      cpu: 1
      memory: 1Gi
```

- Run the following helm upgrade command after updating the values.yaml file.
  `helm upgrade <helm chart> [--version <OCUDR version>] --name <release> --namespace <ocudr-namespace> -f <ocudr_values.yaml>`

Verifying Resource Allocation Issues on Webscale Environment

Webscale environment has openshift container installed. There can be cases where,

- Pods does not scale after you run the installation command and the helm install command fails with timeout error. In this case, check for preinstall hooks failure. Run the **oc get job** command to create the jobs. Describe the job for which the pods are not getting scaled and check if there are **quota limit exceeded** errors with CPU or memory.
- Any of the actual microservice pods do not scale post the hooks completion. In this case, run the **oc get rs** command to get the list of replicaset created for the NF deployment. Then, describe the replicaset for which the pods are not getting scaled and check for **resource quota limit exceeded** errors with CPU or memory.
- Helm install command times-out after all the microservice pods are scaled as expected with the expected number of replicas. In this case, check for post install hooks failure. Run the **oc get job** command to get the post install jobs and do a describe on the job for which the pods are not getting scaled and check if there are **quota limit exceeded** errors with CPU or memory.
- Resource quota exceed beyond limits.

4.3.2.2 Debugging Pod Startup Failure

Follow the guidelines shared below to debug the pod startup failure liveness check issues:

- If dr-service, diameter-proxy, and diam-gateway services are stuck in the CrashLoopBackOff state, then the reason could be that config-server is not yet up. A sample log on these services is as follows:

  ```
  "Config Server is Not yet Up, Wait For config server to be up."
  ```

  To resolve this, make sure the dependent services nudr-config and nudr-config-server is up or the startup probe will attempt to restart pod for every configured amount of time.
- If the notify-service and on-demand migration service is stuck in the Init state, then the reason could be the dr-service is not yet up. A sample log on these services is as follows:

  ```
  "DR Service is Not yet Up, Wait For dr service to be up."
  ```

To resolve this, check for failures on dr-service or the startup probe will attempt to restart pod for every configured amount of time.

- If the microservices connecting to mySQL database is stuck in Crashloopbackoff state, check for mySQL exceptions in the logs and fix accordingly or If you receive error messages `The database secret is empty` or `Invalid data present in the secret` in the main service container logs make sure that the secret is created as mentioned in document and check for the case sensitivity of the keys in the secret. For example, encryptionKey, dsusername, and so on.

4.3.2.3 Debugging UDR with Service Mesh Failure

There are some known failure scenarios that you can encounter while installing UDR with service mesh. The scenarios along with their solutions are as follows:

- **Istio-Proxy side car container not attached to Pod**: This particular failure arises when istio injection is not enabled on the NF installed namespace. Run the following command to verify the same:
  `kubectl get namespace -L istio-injection`

Figure 4-6 Verifying Istio-Proxy

```
[root@master ocudr_1.7.0]# kubectl get namespace -L istio-injection
NAME              STATUS   AGE     ISTIO-INJECTION
default           Active   28d
istio-system      Active   20d
kube-node-lease   Active   28d
kube-public       Active   28d
kube-system       Active   28d
myudr             Active   18d     enabled
myudr1            Active   18d     enabled
occne-infra       Active   27d
ocnrf             Active   20d     enabled
ocudr             Active   26d     disabled
ocudr1            Active   14d
provgw            Active   19d     enabled
vnnrf             Active   4d12h   enabled
```

To enable the istio injection, run the following command:

```
kubectl label --overwrite namespace <nf-namespace> istio-injection=enabled
```

- If any of the hook pods is not responding in the 'NotReady' state and is not cleared after completion, check if the following configuration is set to 'true' under global section. Also, ensure that the URL configured for istioSidecarQuitUrl is correct.

Figure 4-7 When Hook Pod is NotReady

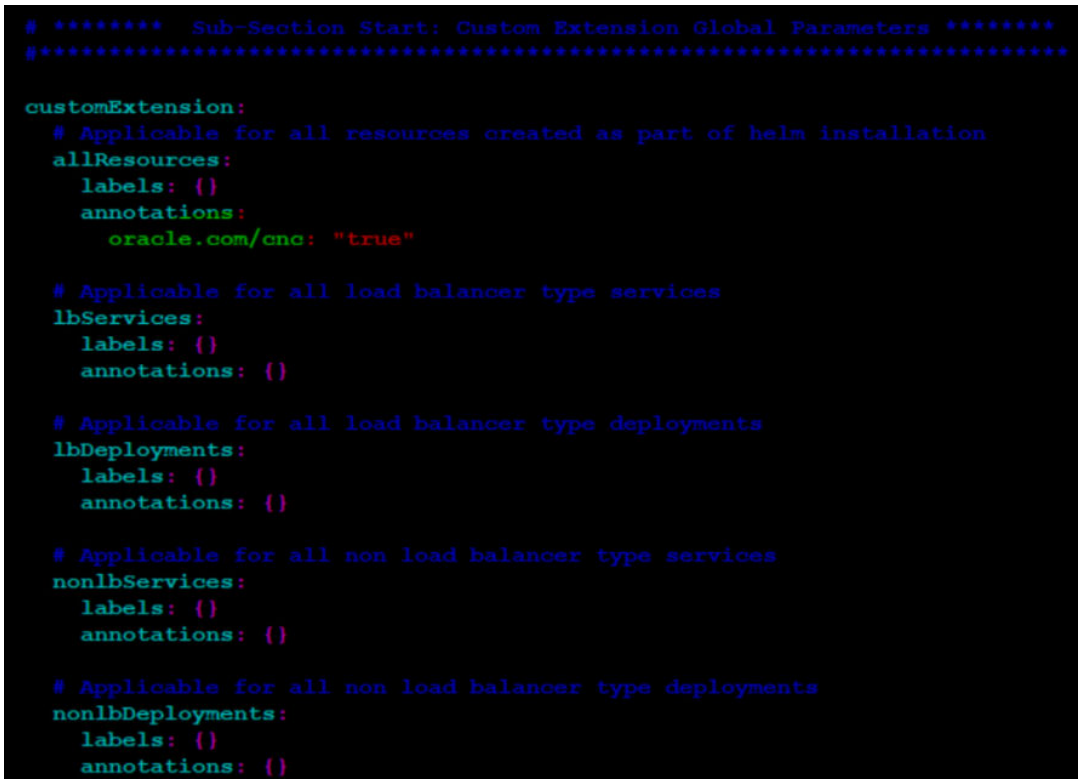
```
# ServiceMesh Related Configurations
# Enable when deployed with ASM
serviceMeshCheck: &serviceMeshFlag true
# Default value: http://127.0.0.1:15020/ready
istioSidecarReadyUrl: &istioReadyUrl "http://127.0.0.1:15000/ready"
#The sidecar (istio url) when deployed in serviceMesh
# Default value: http://127.0.0.1:15020/quitquitquit
istioSidecarQuitUrl: &istioQuitUrl "http://127.0.0.1:15020/quitquitquit"
```

- When Prometheus does not scrape metrics from nudr-nrf-client-service, see if the following annotation is present under nudr-nrf-client-service:

Figure 4-8 nrf-client service

- If there are issues in viewing UDR metrics on OSO Prometheus, you need to ensure that the following highlighted annotation is added to all deployments for the NF.

Figure 4-9 Issues in Viewing UDR Metrics - Add Annotation



- When vDBTier is used as backend and there are connectivity issues, and when nudr-preinstall communicates with DB, which can be seen from error logs on preinstall hook pod, then make the destination rule and service entry for mysql-connectivity-service on occne-infra namespace.
- When installed on ASM, if ingressgateway, egressgateway, or alternate-route services go into CrashLoopBackOff, you must check if the coherence ports is excluded for inbound and outbound on the istio-proxy.
- On the latest F5 versions, if the default istio-proxy resources assigned is less, make sure that you assign minimum one CPU and one GB RAM for all UDR services. The traffic handling services must be same as mentioned in the resource profile. If the pods crash

due to less memory, you must check the configuration. You can refer the following annotations in the custom values file.

```
deployment:
# Replica count for deployment
replicaCount: 2
# Microservice specific notation for deployment
customExtension:
labels: {}
 annotations:
sidecar.istio.io/proxyCPU: "1000m"
sidecar.istio.io/proxyCPULimit: "lOOOm"
sidecar.istio.io/proxyMemory: "lGi"
sidecar.istio.io/proxyMemoryLimit : "lGi"
proxy.istio.io/config: |
terminationDrainDuration: 60s
```

4.3.2.4 Debugging SLF Default Group ID related Issues

SLF default group ID is added to the SLF_GROUP_NAME table through Helm hooks during UDR installation or upgrade. If a subscriber is not found and default group ID is enabled, then a response with default group ID is sent. If the default group ID is not found in the response, then use the following API to add the Default Group ID (This is similar to other SLF Groups PUT operation).

```
http://localhost:8080/slf-group-prov/v1/slf-group
```

```
{
    "slfGroupName": "DefaultGrp",
    "slfGroupType": "LteHss",
    "nfGroupIDs": {
        "NEF": "nef-group-default",
        "UDM": "udm-group-default",
        "PCF": "pcf-group-default",
        "AUSF": "ausf-group-default",
        "CHF": "chf-group-default"
    }
}
```

The default group name is dynamically editable through CNCC. If user changes default group name on CNCC and does not add the same to SLF_GROUP_NAME, then the default group name can be added through API as mentioned above.

4.3.2.5 Debugging Subscriber Activity Logging

This section describes how to troubleshoot the subscriber activity logging related issues.

- If subscriber activity logging is not enabled, check the subscriberAcitivtiyEnabled flag and subscriberIdentifiers keys in the Global Configurations Parameters. For more information, see Oracle Communications Cloud Native Core, Unified Data Repository User Guide.
- If you are not getting the subscriber logs after enabling the flag, then make sure that the subscriber identifiers mentioned in the configuration API contains the same key value as the testing subscriber identifiers.

- Each subscriber identifiers can be configured up to 100 keys using CNC Console or REST API .
- You can remove a subscriber from this feature by removing the subscriber identifiers key from the Global Configurations Parameters as shown below:

```
"subscriberActivityEnabled": true, "subscriberldentifers": {
"nai": [],
"imsi": [
"1111111127,1111111128"
],
"extid": [],
"msisdn": [
"1111111129,1111111130"
]
```

4.3.2.6 Debugging Subscriber Bulk Import Tool Related Issues

Subscriber bulk Import tool pod can run into pending state during installation. The reasons could be as follow:

- Resources is not available. In this case, allocate more resources for the namespace. The "kubectl describe pod" will give you more details on this issue.
- PVC allocation failed for subscriber bulk import tool. During re-installation, there can be a case where the existing PVC is not linked to subscriber bulk import tool. You can debug the issue based on the details from the describe pod output.
- The storage class is not configured correctly. In this case, check the correctness of the configuration as below.

Figure 4-10 Bulk Import Persistent Claim

```
createBulkImportPVC:
  name: bulkimportpersistentclaim
  storageClassName: standard
```

- When subscriber bulk tool installation is complete, there can be a case where REST APIs configurations is not working. In this case, make sure that the below configuration is updated in the custom-values.yaml file.

Figure 4-11 OCUDR Release name

```
# Release Name of OCUDR, To form config-server/ocudr endpoint fqdn
ocudrReleaseName: "ocudr"
```

- If the transfer-in and transfer-out functionality are not working after the remote host transfer is enabled. Make sure that the below steps are performed to resolve the issue:
  - The kubernetes secrets created are correct for the private and public keys.
  - The remote host configured and secrets created are of the same remote host.

- The remote path is correct.
- The space remaining in the remote host is within the limits of the file size that you are transferring.

4.3.2.7 Debugging NF Scoring for a Site

If there are issues related to NF Scoring, then perform the following steps:

- Perform a GET request using `http://<nudr-config-host>:<nudr-config-port>/udr/nf-common-component/v1/app-info/nfScoring` to check if the NF Scoring feature is enabled. If the feature is disabled, the request will show an "ERROR feature not enabled". To enable the feature, use the above GET API to set the feature flag to true and then fetch the NF score.
- To get the detailed information on provisioning and signaling, multiple ingress gateways must be set to true for UDR and SLF.
- If the `Custom Criteria` is enabled and the calculation of NF score for custom criteria fails, you must check the name of the metric and other configured details in custom criteria.

4.3.2.8 Debugging Subscriber Export Tool Related Issues

Subscriber export tool pod can run into pending state during installation. The reasons could be as follow:

- Resources is not available. In this case, allocate more resources for the namespace. The "kubectl describe pod" will give you more details on this issue.
- PVC allocation failed for subscriber bulk import tool. During re-installation, there can be a case where the existing PVC is not linked to subscriber export tool. You can debug the issue based on the details from the describe pod output.
- The storage class is not configured correctly. In this case, check the correctness of the configuration as below.

Figure 4-12 Export tool Presistent Claim

```
createExportToolPVC:
  name: exporttoolpersistentclaim
  storageClassName: standard
```

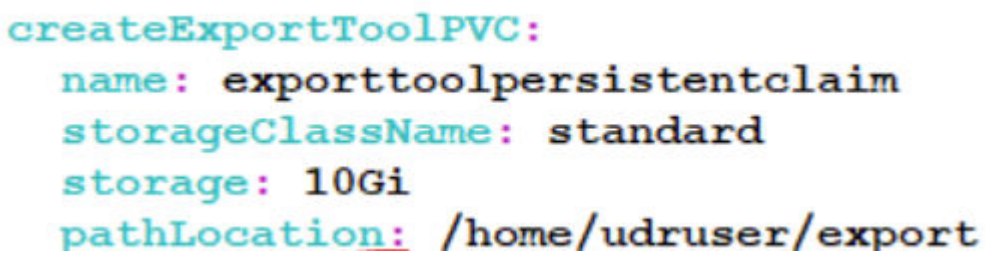
- When subscriber export tool installation is complete, there can be a case where the REST APIs configuration is not working. In this case, make sure that the below configuration is updated in the custom-values.yaml file.

Figure 4-13 OCUDR Release name

```
# Release Name of OCUDR, To form config-server/ocudr endpoint fqdn
ocudrReleaseName: "ocudr"
```

- If the export dump is not generated, you can check the logs for more details. Check the configuration is updated correctly as below.

Figure 4-14 Export Tool Persistent Claim Standard



- If the transfer-in and transfer-out functionality are not working after the remote host transfer is enabled. Make sure that the below steps are performed to resolve the issue:
  - The kubernetes secrets created are correct for the private and public keys.
  - The remote host configured and secrets created are of the same remote host.
  - The remote path is correct.
  - The space remaining in the remote host is within the limits of the file size that you are transferring.

4.3.2.9 Debugging Controlled Shutdown Related Issues

If there are issues related to controlled shutdown, then perform the following steps:

- Check the REST API GLOBAL configuration section if the control shutdown is not enabled. Make sure the flag `enableControlledShutdown` parameter is set to true to enable the feature.
- Once the flag is enabled to true, you can do a PUT request to `udr/nf-common-component/v1/operationalState`. The PUT request throws an error if the flag is disabled.
- When the operational state is set to COMPLETE_SHUTDOWN, all the ingress gateway requests are rejected with the configured error codes. If the request is not rejected, check if the feature flag is enabled and do a GET request on `udr/nf-common-component/v1/operationalState`.
- The subscriber export tool and subscriber import tool rejects all the new request that is queued for processing.
- When the operational state is COMPLETE_SHUTDOWN the NF status is updated as SUSPENDED at NRF. Check the app-info logs if the status is not updated to SUSPENDED. The logs contain the operational state of COMPLETE_SHUTDOWN.

4.3.2.10 Debug Readiness Failure

During the lifecycle of a pod if the pod containers is in NotReady state the reasons could be as follow:

- Make sure that the dependent services is up. Check the logs for the below content:
  `Dependent services down, Set readiness state to REFUSING_TRAFFIC`
- Make sure that the database is available and the app-info is ready to monitor the database. Check the logs for the below content:
  `DB connection down, Set readiness state to REFUSING_TRAFFIC`
- The readiness failure can occur, if resource map or key map table in the database are not having proper content. Check the logs for the below content:
  `ReourceMap/KeyMap Entries missing, Set readiness state to REFUSING_TRAFFIC`

4.3.2.11 Enable cnDBTier Metrics with OSO Prometheus

cnDBTier setup must be applied with the below yaml file. For example:

```
kubectl create -f <.yaml> -n <nsdbtier>
```

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: cndb-to-mysql-external-se
spec:
  exportTo:
  - "."
  hosts:
  - mysql-connectivity-service
  location: MESH_EXTERNAL
  ports:
  - number: 3306
    name: mysql2
    protocol: MySQL
---
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: nf-to-nf
spec:
  exportTo:
  - "."
  hosts:
  - "*.$DOMAIN_NAME"   # DOMAIN_NAME must be replaced with the deployed CNE
Domain name
  location: MESH_EXTERNAL
  ports:
  - number: 80
    name: HTTP2-80
    protocol: TCP
  - number: 8080
    name: HTTP2-8080
    protocol: TCP
  - number: 3306
    name: TCP-3306
    protocol: TCP
  - number: 1186
    name: TCP-1186
    protocol: TCP
  - number: 2202
    name: TCP-2202
    protocol: TCP
  resolution: NONE
---
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
```

```
spec:
  mtls:
    mode: PERMISSIVE
---
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: nf-to-kube-api-server
spec:
  hosts:
  - kubernetes.default.svc.$DOMAIN_NAME  # DOMAIN_NAME must be replaced with
the deployed CNE Domain name
  exportTo:
  - "."
  addresses:
  - 172.16.13.4
  location: MESH_INTERNAL
  ports:
  - number: 443
    name: https
    protocol: HTTPS
  - number: 6443
    name: https-1
    protocol: HTTPS
  resolution: NONE
---
```

Install Operations Services Overlay (OSO) Promethues with cnDBTier namespace in the OSO custom-values.yaml file to remove the cnDBTier metrics. The sample yaml file is given below:

```
##############################################################################
####
#
   #
# Copyright (c) 2022 Oracle and/or its affiliates. All rights
reserved.          #
#
   #
##############################################################################
####

nameOverride: prom

## Helm-Test (Optional)
# Values needed for helm-test, Comment the entire Section if Helm-test not
needed.
helmtestimage: occne-repo-host:5000/k8s.gcr.io/ingress-nginx/controller:v1.3.1
useasm: false
namespace: ocudr-ns
clustername: cne-23-1-rc2
resources:
  limits:
    cpu: 10m
    memory: 32Mi
  requests:
```

```
    cpu: 10m
    memory: 32Mi
promsvcname: oso-prom-svr
almsvcname: oso-prom-alm
prometheushealthyurl: /prometheus/-/healthy
prometheusreadyurl: /prometheus/-/ready

# Note: There are 3 types of label definitons provided in this custom values
file
# TYPE1: Global(allResources)
# TYPE2: lb & nonlb TYPE label only
# TYPE3: service specific label
## NOTE: POD level labels can be inserted using the specific pod label
sections, every pod/container has this label defined below in all components
sections.
# ******** Custom Extension Global Parameters ********
#*************************************************************************

global_oso:
# Prefix & Suffix that will be added to containers
  k8Resource:
    container:
      prefix:
      suffix:

# Service account for Prometheus, Alertmanagers
  serviceAccountNamePromSvr: ""
  serviceAccountNameAlertMgr: ""

  customExtension:
# TYPE1 Label
    allResources:
      labels: {}

# TYPE2 Labels
    lbServices:
      labels: {}

    nonlbServices:
      labels: {}

    lbDeployments:
      labels: {}

    nonlbDeployments:
      labels: {}

    lbStatefulSets:
      labels: {}

# Add annotations for disabling sidecar injections into oso pods here
# eg: annotations:
#       - sidecar.istio.io/inject: "false"
annotations:
  - sidecar.istio.io/inject: "false"
```

```
## Setting this parameter to false will disable creation of all default
clusterrole, clusterolebing, role, rolebindings for the componenets that are
packaged in this csar.
rbac:
  create: true

podSecurityPolicy:
  enabled: false

## Define serviceAccount names for components. Defaults to component's fully
qualified name.
##
serviceAccounts:
  alertmanager:
    create: true
    name:
    annotations: {}
  nodeExporter:
    create: false
    name:
    annotations: {}
  pushgateway:
    create: false
    name:
    annotations: {}
  server:
    create: true
    name:
    annotations: {}

alertmanager:
  enabled: true

  ## Use a ClusterRole (and ClusterRoleBinding)
  ## - If set to false - Define a Role and RoleBinding in the defined
namespaces ONLY
  ## This makes alertmanager work - for users who do not have ClusterAdmin
privs, but wants alertmanager to operate on their own namespaces, instead of
clusterwide.
  useClusterRole: false

  ## Set to a rolename to use existing role - skipping role creating - but
still doing serviceaccount and rolebinding to the rolename set here.
  useExistingRole: false

  ## alertmanager resources name
  name: alm
  image:
    repository: occne-repo-host:5000/quay.io/prometheus/alertmanager
    tag: v0.24.0
    pullPolicy: IfNotPresent
  extraArgs:
    data.retention: 120h
  prefixURL: /cne-23-1-rc2/alertmanager
  baseURL: "http://localhost/cne-23-1-rc2/alertmanager"
  configFileName: alertmanager.yml
```

```
  nodeSelector: {}
  affinity: {}
  podDisruptionBudget:
    enabled: true
    minAvailable: 1
  persistentVolume:
    enabled: true
    accessModes:
      - ReadWriteOnce
    annotations: {}
    ## alertmanager data Persistent Volume existing claim name
    ## Requires alertmanager.persistentVolume.enabled: true
    ## If defined, PVC must be created manually before volume will be bound
    existingClaim: ""
    mountPath: /data
    size: 2Gi
    storageClass: "standard"

  ## Annotations to be added to alertmanager pods
  ##
  podAnnotations: {}

  ## Labels to be added to Prometheus AlertManager pods
  ##
  podLabels: {}

  replicaCount: 2

  ## Annotations to be added to deployment
  ##
  deploymentAnnotations: {}

  statefulSet:
    ## If true, use a statefulset instead of a deployment for pod management.
    ## This allows to scale replicas to more than 1 pod
    ##
    enabled: true
    annotations: {}
    labels: {}
    podManagementPolicy: OrderedReady

    ## Alertmanager headless service to use for the statefulset
    ##
    headless:
      annotations: {}
      labels: {}

      ## Enabling peer mesh service end points for enabling the HA alert
manager
      ## Ref: https://github.com/prometheus/alertmanager/blob/master/README.md
      enableMeshPeer: true

      servicePort: 80

  ## alertmanager resource requests and limits
  ## Ref: http://kubernetes.io/docs/user-guide/compute-resources/
```

```
  ##
  resources:
    limits:
      cpu: 20m
      memory: 64Mi
    requests:
      cpu: 20m
      memory: 64Mi

  service:
    annotations: {}
    labels: {}
    clusterIP: ""
    loadBalancerIP: ""
    loadBalancerSourceRanges: []
    servicePort: 80
    # nodePort: 30000
    sessionAffinity: None
    type: ClusterIP

## Monitors ConfigMap changes and POSTs to a URL
## Ref: https://github.com/jimmidyson/configmap-reload
##
configmapReload:
  prometheus:
    enabled: true
    ## configmap-reload container name
    ##
    name: configmap-reload
    image:
      repository: occne-repo-host:5000/docker.io/jimmidyson/configmap-reload
      tag: v0.8.0
      pullPolicy: IfNotPresent

    # containerPort: 9533

    ## Additional configmap-reload mounts
    ##
    extraConfigmapMounts: []

    ## Security context to be added to configmap-reload container
    containerSecurityContext: {}

    ## configmap-reload resource requests and limits
    ## Ref: http://kubernetes.io/docs/user-guide/compute-resources/
    ##
    resources:
      limits:
        cpu: 10m
        memory: 32Mi
      requests:
        cpu: 10m
        memory: 32Mi

  alertmanager:
    enabled: true
```

```
    name: configmap-reload
    image:
      repository: occne-repo-host:5000/docker.io/jimmidyson/configmap-reload
      tag: v0.8.0
      pullPolicy: IfNotPresent

    # containerPort: 9533
    ## Additional configmap-reload mounts
    ##
    extraConfigmapMounts: []
      # - name: prometheus-alerts
      #   mountPath: /etc/alerts.d
      #   subPath: ""
      #   configMap: prometheus-alerts
      #   readOnly: true

    resources:
      limits:
        cpu: 10m
        memory: 32Mi
      requests:
        cpu: 10m
        memory: 32Mi

kubeStateMetrics:
  enabled: false

nodeExporter:
  enabled: false

server:
  enabled: true
  ## namespaces to monitor (instead of monitoring all - clusterwide). Needed
if you want to run without Cluster-admin privileges.
  namespaces: []
  #  - ocudr-ns
  name: svr
  image:
    repository: occne-repo-host:5000/quay.io/prometheus/prometheus
    tag: v2.39.1
    pullPolicy: IfNotPresent
  prefixURL: /cne-23-1-rc2/prometheus
  baseURL: "http://localhost/cne-23-1-rc2/prometheus"

  ## Additional server container environment variables
  env: []

  # List of flags to override default parameters, e.g:
  # - --enable-feature=agent
  # - --storage.agent.retention.max-time=30m
  defaultFlagsOverride: []

  extraFlags:
    - web.enable-lifecycle
    ## web.enable-admin-api flag controls access to the administrative HTTP
API which includes functionality such as
```

```
    ## deleting time series. This is disabled by default.
    # - web.enable-admin-api
    ##
    ## storage.tsdb.no-lockfile flag controls BD locking
    # - storage.tsdb.no-lockfile
    ##
    ## storage.tsdb.wal-compression flag enables compression of the write-
ahead log (WAL)
    # - storage.tsdb.wal-compression

  ## Path to a configuration file on prometheus server container FS
  configPath: /etc/config/prometheus.yml

  global:
    scrape_interval: 1m
    scrape_timeout: 30s
    evaluation_interval: 1m
  #remoteWrite:
    #- url OSO_CORTEX_URL
    # remote_timout (default = 30s)
      #remote_timeout: OSO_REMOTE_WRITE_TIMEOUT
    # bearer_token for cortex server to be configured
    #      bearer_token: BEARER_TOKEN

  extraArgs:
    storage.tsdb.retention.size: 1GB
  ## Additional Prometheus server Volume mounts
  ##
  extraVolumeMounts: []

  ## Additional Prometheus server Volumes
  ##
  extraVolumes: []

  ## Additional Prometheus server hostPath mounts
  ##
  extraHostPathMounts: []
    # - name: certs-dir
    #   mountPath: /etc/kubernetes/certs
    #   subPath: ""
    #   hostPath: /etc/kubernetes/certs
    #   readOnly: true

  extraConfigmapMounts: []

  nodeSelector: {}
  affinity: {}
  podDisruptionBudget:
    enabled: false
    maxUnavailable: 1

  persistentVolume:
    enabled: true
    accessModes:
      - ReadWriteOnce
    annotations: {}
```

```
  ## Prometheus server data Persistent Volume existing claim name
  ## Requires server.persistentVolume.enabled: true
  ## If defined, PVC must be created manually before volume will be bound
  existingClaim: ""
  size: 2Gi
  storageClass: "standard"

## Annotations to be added to Prometheus server pods
##
podAnnotations: {}

## Labels to be added to Prometheus server pods
##
podLabels: {}

## Prometheus AlertManager configuration
##
alertmanagers:
- kubernetes_sd_configs:
    - role: pod
  # Namespace to be configured
      namespaces:
        names:
        - ocudr-ns
        - dbtier-ns
  path_prefix: cne-23-1-rc2/alertmanager
  relabel_configs:
  - source_labels: [__meta_kubernetes_namespace]
  # namespace to be configured
    regex: ocudr-ns
    action: keep
  - source_labels: [__meta_kubernetes_pod_label_app]
    regex: prom
    action: keep
  - source_labels: [__meta_kubernetes_pod_label_component]
    regex: alm
    action: keep
  - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_probe]
    regex: .*
    action: keep
  - source_labels: [__meta_kubernetes_pod_container_port_number]
    regex:
    action: drop

## Use a StatefulSet if replicaCount needs to be greater than 1 (see below)
##
replicaCount: 1

## Annotations to be added to deployment
##
deploymentAnnotations: {}

statefulSet:
  ## If true, use a statefulset instead of a deployment for pod management.
  ## This allows to scale replicas to more than 1 pod
```

```
    ##
    enabled: false

    annotations: {}
    labels: {}
    podManagementPolicy: OrderedReady

  resources:
    limits:
      cpu: 2
      memory: 4Gi
    requests:
      cpu: 2
      memory: 4Gi

  service:
    enabled: true
    annotations: {}
    labels: {}
    clusterIP: ""
    externalIPs: []
    loadBalancerIP: ""
    loadBalancerSourceRanges: []
    servicePort: 80
    sessionAffinity: None
    type: NodePort

    ## If using a statefulSet (statefulSet.enabled=true), configure the
    ## service to connect to a specific replica to have a consistent view
    ## of the data.
    statefulsetReplica:
      enabled: false
      replica: 0
  retention: "7d"

pushgateway:
  ## If false, pushgateway will not be installed
  ##
  enabled: false

## alertmanager ConfigMap entries
##
alertmanagerFiles:
  alertmanager.yml:
    global: {}
      # slack_api_url: ''

    receivers:
      - name: default-receiver
        # slack_configs:
        #  - channel: '@you'
        #    send_resolved: true

    route:
      group_wait: 10s
      group_interval: 5m
```

```
      receiver: default-receiver
      repeat_interval: 3h

## Prometheus server ConfigMap entries for rule files (allow prometheus
labels interpolation)
ruleFiles: {}

## Prometheus server ConfigMap entries
##
serverFiles:

  ## Alerts configuration
  ## Ref: https://prometheus.io/docs/prometheus/latest/configuration/
alerting_rules/
  alerting_rules.yml: {}

  ## DEPRECATED DEFAULT VALUE, unless explicitly naming your files, please
use alerting_rules.yml
  alerts: {}

  ## Records configuration
  ## Ref: https://prometheus.io/docs/prometheus/latest/configuration/
recording_rules/
  recording_rules.yml: {}
  ## DEPRECATED DEFAULT VALUE, unless explicitly naming your files, please
use recording_rules.yml
  rules: {}

  prometheus.yml:
    rule_files:
      - /etc/config/recording_rules.yml
      - /etc/config/alerting_rules.yml
    ## Below two files are DEPRECATED will be removed from this default
values file
      - /etc/config/rules
      - /etc/config/alerts

    scrape_configs:
      - job_name: prometheus
        metrics_path: cne-23-1-rc2/prometheus/metrics
        static_configs:
          - targets:
            - localhost:9090

extraScrapeConfigs: |

  - job_name: 'oracle-cnc-service'
    kubernetes_sd_configs:
      - role: service
        namespaces:
          names:
          - ocudr-ns
          - dbtier-ns
        #  - ns2
    relabel_configs:
      - source_labels: [__meta_kubernetes_service_annotation_oracle_com_cnc]
```

```
        regex: true
        action: keep
      - source_labels:
[__meta_kubernetes_service_annotation_prometheus_io_scrape]
        action: keep
        regex: true
      - source_labels:
[__meta_kubernetes_service_annotation_prometheus_io_path]
        action: replace
        target_label: __metrics_path__
        regex: (.+)
      - source_labels: [__address__,
__meta_kubernetes_service_annotation_prometheus_io_port]
        action: replace
        regex: ([^:]+)(?::\d+)?;(\d+)
        replacement: $1:$2
        target_label: __address__
      - action: labelmap
        regex: __meta_kubernetes_service_label_(.+)
      - source_labels: [__meta_kubernetes_namespace]
        action: replace
        target_label: kubernetes_namespace
      - source_labels: [__meta_kubernetes_service_name]
        action: replace
        target_label: kubernetes_service_name

  - job_name: 'oracle-cnc-pod'
    kubernetes_sd_configs:
      - role: pod
        namespaces:
          names:
          - ocudr-ns
          - dbtier-ns
        #  - ns2
    relabel_configs:
      - source_labels: [__meta_kubernetes_pod_annotation_oracle_com_cnc]
        regex: true
        action: keep
      - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_scrape]
        action: keep
        regex: true
      - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_path]
        action: replace
        target_label: __metrics_path__
        regex: (.+)
      - source_labels: [__address__,
__meta_kubernetes_pod_annotation_prometheus_io_port]
        action: replace
        regex: ([^:]+)(?::\d+)?;(\d+)
        replacement: $1:$2
        target_label: __address__
      - action: labelmap
        regex: __meta_kubernetes_pod_label_(.+)
      - source_labels: [__meta_kubernetes_namespace]
        action: replace
        target_label: kubernetes_namespace
```

```
      - source_labels: [__meta_kubernetes_pod_name]
        action: replace
        target_label: kubernetes_pod_name

  - job_name: 'oracle-cnc-endpoints'
    kubernetes_sd_configs:
      - role: endpoints
        #namespaces:
        #  names:
        #  - ns1
        #  - ns2
    relabel_configs:
      - source_labels: [__meta_kubernetes_service_annotation_oracle_com_cnc]
        regex: true
        action: keep
      - source_labels:
[__meta_kubernetes_service_annotation_prometheus_io_scrape]
        action: keep
        regex: true
      - source_labels:
[__meta_kubernetes_service_annotation_prometheus_io_scheme]
        action: replace
        target_label: __scheme__
        regex: (https?)
      - source_labels:
[__meta_kubernetes_service_annotation_prometheus_io_path]
        action: replace
        target_label: __metrics_path__
        regex: (.+)
      - source_labels: [__address__,
__meta_kubernetes_service_annotation_prometheus_io_port]
        action: replace
        target_label: __address__
        regex: ([^:]+)(?::\d+)?;(\d+)
        replacement: $1:$2

  - job_name: 'oracle-cnc-ingress'
    kubernetes_sd_configs:
      - role: ingress
        #namespaces:
        #  names:
        #  - ns1
        #  - ns2
    relabel_configs:
      - source_labels: [__meta_kubernetes_ingress_annotation_oracle_com_cnc]
        regex: true
        action: keep
```

4.3.2.12 Debugging ndbmysqld Pods Restart During cnDBTier Installation or Upgrade

During cnDBTier Installation or Upgrade the readiness probe fails as the ndbmysqld pods wait for the data nodes to be up and running. This causes the ndbmysqld pods to restart with `Reason: Error and Exit Code: 1` error. If the data nodes takes time to come up for any reason, such as slowness of cluster, the ndbmysqld pods will restart. The ndbmysqld pods stabilize when the data nodes comes up.

4.3.2.13 Debugging Error Logging Related Issues

If there are issues related to Error Logging, then perform the following steps:

- You must set the `additionalErrorLogging` parameter to ENABLED per microservice for the Error Logging feature to work. This feature is "DISABLED" by default and it can be ENABLED or DISABLED using REST APIs, CNC console, or by changing the values in custom values yaml file during installation.
- For logging subscriber information in the logs, you must set the `logSubscriberInfo` parameter to "ENABLED" per microservice. The parameter can be ENABLED or DISABLED using REST APIs, CNC console, or by changing the values in custom values yaml file during installation.

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

4.3.2.14 Debugging Suppress Notification Related Issues

If there are issues related to Suppress Notification, then perform the following steps:

- You must set the `suppressNotificationEnabled` parameter to true in the global section of the custom values yaml file for the Suppress Notification feature to work. This feature is enabled by default and it can be enabled or disabled using REST APIs, CNC console, or by changing the values in custom values yaml file during installation.
- If you observe unexpected notification then check if the feature is enabled from the global configuration using configuration REST APIs.
- If the feature is enabled and you observe unexpected notifications for update requests then compare the User-Agent received in the request header with the User-Agent received in the subscription request.
- This feature is applicable only for signaling requests. For provisioning request the notification generation behavior remains same as earlier.
- This feature does not work with the subscription created in the previous release versions. You must create new subscriptions with the feature enabled for Suppress Notification feature to work.

```
#This flag to enable suppress notification feature
suppressNotificationEnabled: true
```

4.3.2.15 Debugging Diameter S13 Interface Related Issues

If there are issues related to Diameter S13 Interface, then perform the following steps:

- If `global.s13InterfaceEnable` flag is set to true and if the helm installation is throwing errors, you must enable the following parameters in `ocudr-custom-values.yaml` file:
  - `global.diamGatewayEnable`
  - `nudr-diameterproxy.enabled`
- If the NRF client heartbeats do not consider the diameter services when diameter S13 interface is enabled, you must check the following configuration in the `ocudr-custom-values.yaml` file.

> **Note**
>
> The NRF client does not register the EIR with NRF if diameter services is down.

```
#eir mode
eir: &eir
- '{{ .Release.Name }}-nudr-drservice'
- '{{ .Release.Name }}-egressgateway'
- '{{ .Release.Name }}-ingressgateway-sig'
- '{{ .Release.Name }}-ingressgateway-prov'
- '{{ .Release.Name }}-nudr-config' #uncomment only if config service
enabled
- '{{ .Release.Name }}-nudr-config-server' #uncomment only if config
service enabled
- '{{ .Release.Name }}-alternate-route' #uncomment if alternate route
enabled
- '{{ .Release.Name }}-nudr-dr-provservice' # uncomment only if
drProvisioningEnabled is enabled
- '{{ .Release.Name }}-nudr-diameterproxy' # uncomment only if
s13InterfaceEnable is enabled
- '{{ .Release.Name }}-nudr-diam-gateway' # uncomment only if
s13InterfaceEnable is enabled
```

- If the diameter gateway is answering CEA message with `DIAMETER_UNKNOWN_PEER`, then client peer configuration is incorrect. You must perform the configuration in `allowedClientNodes` section of diameter gateway service configuration using REST API for the client to connect to EIR and send an ECR request.
- If the diameter gateway is answering CEA message as success and other diameter message responds with `DIAMETER_UNABLE_TO_COMPLY/DIAMETER_MISSING_AVP`, then the issue could be in the diameter message request.
- If there are error logs in diameter gateway microservice stating that the connection is refused with IP and port numbers, then the specified configured peer node was not able to accept CER request from diameter gateway. The diameter gateway retries multiple times to connect that peer.
- If you are getting `DIAMETER_UNABLE_TO_DELIVERY` error message, then diameterproxy microservice is down.
- If the diam-gateway goes to crashloop back off state, then it could be due to incorrect peer node configuration.
- Active connections to the existing peer nodes can be verified using `ocudr_diam_conn_network` metric.

4.3.2.16 Debugging TLS Related Issues

This section describes the TLS related issues and their resolution steps. It is recommended to attempt the resolution steps provided in this guide before contacting Oracle Support.

Problem: Handshake is not established between UDRs.

Scenario: When the client version is TLSv1.2 and the server version is TLSv1.3.

Server Error Message

```
The client supported protocol versions[TLSv1.2] are not accepted by server
preferences [TLSv1.3]
```

Client Error Message

```
Received fatal alert: protocol_version
```

Scenario: When the client version is TLSv1.3 and the server version is TLSv1.2.

Server Error Message

```
The client supported protocol versions[TLSv1.3]are not accepted by server
preferences [TLSv1.2]
```

Client Error Message

```
Received fatal alert: protocol_version
```

Solution:

If the error logs have the SSL exception, do the following:

Check the TLS version of both UDRs, if both support different and single TLS versions, (that is, UDR 1 supports TLS 1.2 only and UDR 2 supports TLS 1.3 only or vice versa), handshake fails. Ensure that the TLS version is same for both UDRs or revert to default configuration for both UDRs. The TLS version communication supported are:

Table 4-1 TLS Version Used

Client TLS Version	Server TLS Version	TLS Version Used
TLSv1.2, TLSv1.3	TLSv1.2, TLSv1.3	TLSv1.3
TLSv1.3	TLSv1.3	TLSv1.3
TLSv1.3	TLSv1.2, TLSv1.3	TLSv1.3
TLSv1.2, TLSv1.3	TLSv1.3	TLSv1.3
TLSv1.2	TLSv1.2, TLSv1.3	TLSv1.2
TLSv1.2, TLSv1.3	TLSv1.2	TLSv1.2

Check the cipher suites being supported by both UDRs, it should be either the same or should have common cipher suites present. If not, revert to default configuration.

Problem: Pods are not coming up after populating the `clientDisabledExtension` or `serverDisabledExtension` Helm parameter.

Solution:

- Check the value of the `clientDisabledExtension` or `serverDisabledExtension` parameters. The following extensions should not be present for these parameters:
  - supported_versions
  - key_share
  - supported_groups
  - signature_algorithms
  - pre_shared_key

If any of the above values is present, remove them or revert to default configuration for the pod to come up.

Problem: Pods are not coming up after populating the `clientSignatureSchemes` Helm parameter.

Solution:

- Check the value of the `clientSignatureSchemes` parameter.
- The following values should be present for this parameter:
  - rsa_pkcs1_sha512
  - rsa_pkcs1_sha384
  - rsa_pkcs1_sha256

  If any of the above values is not present, add them or revert to default configuration for the pod to come up.

4.3.2.17 Debugging Dual Stack Related Issues

With this feature, cnUDR can be deployed on a dual stack Kubernetes infrastructure. Using the dual stack mechanism, cnUDR establishes and accepts connections within the pods and services in a Kubernetes cluster using IPv4 or IPv6. 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` and the values can be changed in `ocudr-custom values.yaml` file during installation. The Helm configuration is as follows:

- To use this feature, cnUDR must be deployed on a dual stack Kubernetes infrastructure either in IPv4 preferred CNE or IPv6 preferred CNE.
- If the `global.deploymentMode` parameter is set to 'IPv6_IPv4' then, when all the pods are running, the services, such as ingressgateway-prov and ingressgateway-sig must have both IPv6 and IPv4 addresses assigned. The default address must be IPv6 . The IP family policy must be set to `RequireDualStack`. The load balancer assigned must have both IPv4 and IPv6 addresses.
- All internal services must be single stack and must have only IPv6 and IP family policy. All the pods must have both IPv4 and IPv6 addresses.

- This feature does not work after upgrade since the upgrade path is not identified for this feature. The operators must perform a fresh installation of the NF to enable the Dual Stack functionality.

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

4.3.2.18 Debugging Lifecycle Management (LCM) Based Automation Issues

Perform the following steps if there are issues related to Lifecycle Management (LCM) Based Automation feature:

- Make sure that the `autoCreateResources.enabled` and `autoCreateResources.serviceaccounts.enabled` flags are enabled.
- During upgrade, if a new service account name is provided in the `serviceAccountName` parameter with `autoCreateResources.enabled` and `autoCreateResources.serviceaccounts.enabled` flags enabled, then a new service account name is created. If the service account name is not created, then you must check the configuration and the flags again.
- During upgrade, you must use a different service account name when you upgrading from manual to automation. If you use same service account name when upgrading from manual to automation, then Helm does not allow upgrade due to ownership issues.
- To use the OSO alerts automation feature, you must follow the installation steps of the `oso-alr-config` Helm chart by providing the alert file that needs to be applied to the namespace. For more information, see *Oracle Communications Cloud Native Core, Operations Services Overlay Installation and Upgrade Guide*. If the alert file is not provided during Helm installation, you can provide the alert file during upgrade procedure. Alert file can be applied to the namespace during Helm installation or upgrade procedure.
- If an incorrect data is added to the alert file, you can clear the entire data in the alert file by providing the empty alert file (`ocslf_alertrules_empty_<version>.yaml`). For more information, see "OSO Alerts Automation" section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.
- If you provide the service account name during the upgrade but the feature is disabled, the nudr-pre-upgrade hook fails because it cannot find the service account. If the upgrade fails, the rollback to the previous version will be unsuccessful due to the missing alternate route service account, resulting in an error message indicating that the service account for the alternate route service is not found. To address this issue, it is necessary to manually create the service account after the initial upgrade failure, then continue with the upgrade, this will also ensures a successful rollback.

4.3.2.19 Troubleshooting TLS 1.3 Support for Kubernetes API Server

If you enable the TLS 1.3 Support for Kubernetes API Server feature by setting `tlsEnableForKubeApiServer` to true and if there is a configuration mismatch, the Helm installation and upgrade fails. For example, if you configure `global.tlsVersionForKubeApiServer` and `global.cipherSuitesForKubeApiServer` incorrectly, as shown in the following example:

```
tlsEnableForKubeApiServer: &tlsEnableForKubeApiServer true
tlsVersionForKubeApiServer: &tlsVersionForKubeApiServer TLSv1.1
cipherSuitesForKubeApiServer: &cipherSuitesForKubeApiServer
```

```
  - TLS_AES_256_GCM_SHA384
  - TLS_AES_128_GCM_SHA256
  - TLS_CHACHA20_POLY1305_SHA256
featureSecretsForKubeApiServer: &featureSecretsForKubeApiServer
  - ocudr-gateway-secret
```

The Helm installation and upgrade fails with the following error:

```
Error: INSTALLATION FAILED: execution error at (ocudr/charts/ingressgateway-
sig/templates/gateway.yml:181:28): Invalid ciphers configured for the
configured kube-api-server tls version
```

4.3.3 Debugging Post Installation Related Issues

This section describes how to troubleshoot the post installation related issues.

4.3.3.1 Debugging Helm Test Issues

To debug the Helm Test issues:

- Run the following command to get the Helm Test pod name.
  `kubectl get pods -n <deployment-namespace>`
- Check for the Helm Test pod that is in error state.

Figure 4-15 Helm Test Pod

```
[root@master ~]# kubectl get pods -n ocudr
NAME                                        READY   STATUS    RESTARTS   AGE
ocudr-egressgateway-595d796-n99r9           1/1     Running   0          2m7s
ocudr-ingressgateway-74c94967c5-kmcfz       1/1     Running   0          2m7s
ocudr-nudr-config-65d8946986-pm561          1/1     Running   0          2m7s
ocudr-nudr-config-server-5c9fb996c7-nwj7h   1/1     Running   0          2m7s
ocudr-nudr-diameterproxy-6bf67d8d8d-6mlkb   1/1     Running   0          2m7s
ocudr-nudr-drservice-595bf9877d-jg58b       0/1     Pending   0          2m7s
ocudr-nudr-notify-service-65cf544955-dgxgq  1/1     Running   0          2m7s
ocudr-nudr-nrf-client-service-64774d996-6s64s 1/1   Running   0          2m7s
ocudr-test-twjqh                            0/1     Error     0          82s
```

- Run the following command to check the Helm Test pod:
  `kubectl logs <helm_test_pod_name> -n <deployment_namespace>`

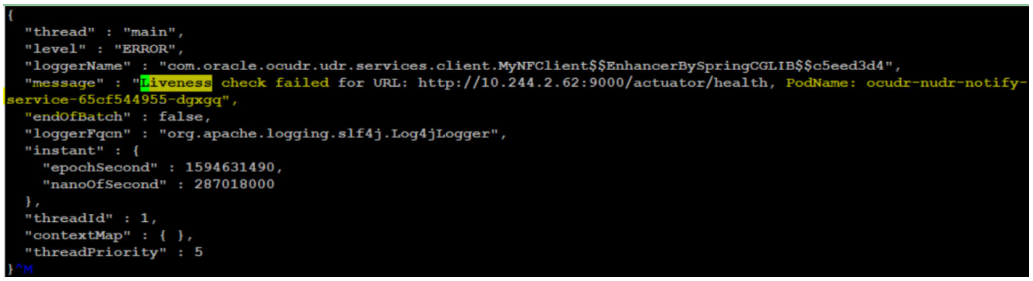
  In the logs, concentrate on ERROR and WARN level logs. There can be multiple reasons for failure. Some of them are shown below:

  – **Figure 4-16 Helm Test in Pending State**

```
{
  "thread" : "main",
  "level" : "ERROR",
  "loggerName" : "com.oracle.ocudr.udr.services.client.MyNFClient$$EnhancerBySpringCGLIB$$c5eed3d4",
  "message" : "Pod check failed, current state: Pending, PodName: ocudr-nudr-drservice-595bf9877d-jg58b",
  "endOfBatch" : false,
  "loggerFqcn" : "org.apache.logging.slf4j.Log4jLogger",
  "instant" : {
    "epochSecond" : 1594631490,
    "nanoOfSecond" : 283784000
  },
  "threadId" : 1,
  "contextMap" : { },
  "threadPriority" : 5
}^M
```

In this case, check for CPU and Memory availability in the Kubernetes cluster.

– **Figure 4-17 Pod Readiness Failed**



In this case, check for the correctness of the readiness probe URL in the particular microservice Helm charts under charts folder. In the above case, check for charts of notify service or check if the pod is crashing for some reason when the URL configured for readiness probe is correct.

– There are a few other cases where the **httpGet** parameter is not configured for the readiness probe. In this case, Helm Test is considered a success for that pod. If the Pod or PVC list is fetched based on namespace and the labelSelector is empty, then the helm test is considered a success.

The Helm test logs generate the following error:

Figure 4-18 Helm Test Log Error

- Check whether the required permission for te resource of the group is missing in the `deployment-rbac.yaml` file. The above sample shows that the permissions are missing for the persistent volume claims.
- Give the appropriate permissions and redeploy.
  Check if the following error appears while running the helm test:

  ```
  14:35:57.732 [main] WARN
  org.springframework.context.annotation.AnnotationConfigApplicationContext
  - Exception encountered during context initialization - cancelling refresh
  attempt: org.springframework.beans.factory.BeanCreationException: Error
  creating bean with name 'k8SFabricClient': Invocation of init method
  failed; nested exception is java.lang.ArrayIndexOutOfBoundsException:
  Index 1 out of bounds for length 1
  { {} }
  ```

- Check the custom values file that is used to create the deployment. The resources should be mentioned in the form of an array under the resources section in the following format: `<k8ResourceName>/<maxNFVersion>`.

4.3.3.2 Debugging Horizontal Pod Autoscaler Issues

There can be scenarios where Horizontal Pod Autoscaler (HPA) running on `nudr-drservice` deployment and `nudr_notify_service` might not get the CPU metrics successfully from the pods. Run the following command to view the HPA details:

```
kubectl get hpa
```

In this scenario, you need to check the following:

- Check whether the metrics server is running on the Kubernetes cluster. If the server is running and the CPU usage pod is still not accessible, check the metrics-server values.yaml file for the arguments passed shown as follows:

Figure 4-19 metrics-server yaml file

```
args:
  - --kubelet-preferred-address-types=InternalIP
  - --kubelet-insecure-tls
```

- If any changes are required, make them, restart the metrics server pod, and check for correctness. Wait a couple of minutes after the metrics server starts to see the CPU usage update on running `kubectl get hpa` command.

Figure 4-20 Debugging HPA Issues

```
[admusr@olmaster templates]$ kubectl get hpa
NAME                        REFERENCE                              TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
ocudr-nudr-drservice        Deployment/ocudr-nudr-drservice        0%/80%    1         1         1          18h
ocudr-nudr-notify-service   Deployment/ocudr-nudr-notify-service   42%/80%   1         1         1          18h
```

4.3.3.3 Debugging HTTPS Support Related Issues

UDR supports HTTPS and its validations at the UDR Ingress Gateway. You may encounter issues related to HTTPS when:

- **HTTPS port is not exposed:** Run the following command to verify if the HTTPS port is exposed:
  `kubectl get svc --n <ocudr-namespace>`

Figure 4-21 HTTPS Port Exposed

```
[root@master ocudr]# kubectl get svc -n ocudr
NAME                              TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)
                  AGE
ocudr-ingressgateway              LoadBalancer   10.102.65.118    <pending>     80:32659/TCP,443:30500/TCP,
5701:30245/TCP    42s
ocudr-nudr-drservice              ClusterIP      None             <none>        5002/TCP,9000/TCP,5001/TCP
                  42s
ocudr-nudr-notify-service         ClusterIP      None             <none>        9000/TCP,5001/TCP,5002/TCP
                  42s
ocudr-nudr-nrf-client-service     ClusterIP      None             <none>        9000/TCP
                  42s
udrdbservice                      ClusterIP      10.111.252.140   <none>        3306/TCP
```

> **Note**
>
> In the above figure, the secure port is 443.

If the HTTPS port is not exposed, then enable the configuration information highlighted in the following figure under the **ingressgateway** section of the values.yaml file.

Figure 4-22 Configuration Info under Ingressgateway

```
#Server Configuration for http and https support
#Server side http support
enableIncomingHttp: true
#Server side https support
enableIncomingHttps: false
#Client side https support
enableOutgoingHttps: false
```

- **Ingress Gateway Container is stuck in Init State/Failed:** The Ingress Gateway container may stop responding due to any one of the following reasons:
  - When config initssl is enabled under ingressgateway section of the values.yaml file.

Figure 4-23 config initssl

```
# To Initialize SSL related infrastructure in init/update container
initssl: false
```

  - If config initssl is enabled, then check whether secrets are created with all required certificates. The following figure shows the commands that you need to run to check whether secrets are present and have all the required data.

Figure 4-24 Commands to Check Secrets

```
[root@master ocudr]# kubectl get secret -n ocudr
NAME                                TYPE                                  DATA   AGE
default-token-g75q7                 kubernetes.io/service-account-token   3      13d
ocudr-secrets                       Opaque                                4      3m35s
ocudr-serviceaccount-token-lwh8k    kubernetes.io/service-account-token   3      2m44s
ocudrgateway-secret                 Opaque                                7      43m
[root@master ocudr]# kubectl describe secret ocudrgateway-secret -n ocudr
Name:         ocudrgateway-secret
Namespace:    ocudr
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
apigatewayecdsa.cer:          1277 bytes
apigatewayrsa.cer:            1554 bytes
caroot.cer:                   1858 bytes
ecdsa_private_key_pkcs8.pem:  241 bytes
key.txt:                      15 bytes
rsa_private_key_pkcs1.pem:    1679 bytes
trust.txt:                    17 bytes
```

- **Config-Server Container not responding in Hooks Init State:** UDR does not respond in the Hooks Init state when there is a database connection failure.

Figure 4-25 Config Server Container Status

In this case, run the describe pod command (on the above pod). In most cases, it is due to secrets not being found.

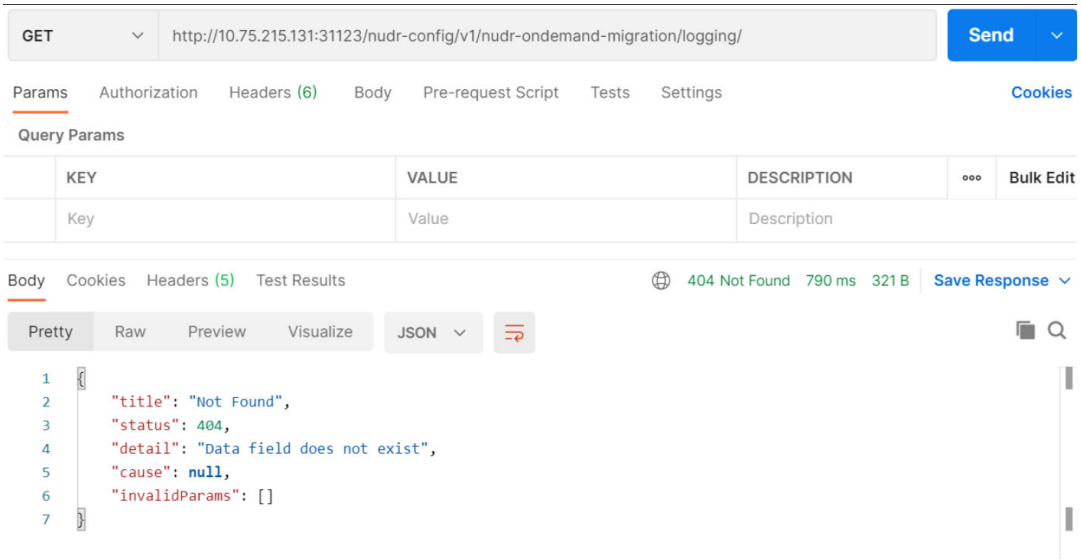
Also, verify the configuration given below to ensure config-server deployment refers to the correct secret values.

```
global:

  dbCredSecretName: 'ocudr-secrets'
```

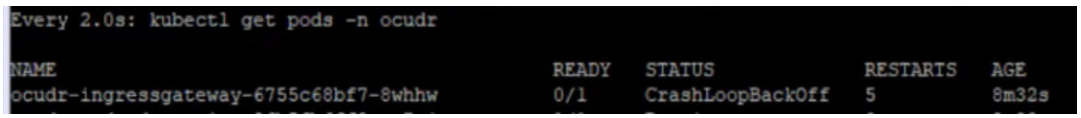
- **Config-Server Post Upgrade Hooks with below error.**:
  When more than one UDR is installed with the same nfInstanceId and in the same udrDB, installation does not cause any issue or error. However, for the second UDR, there are no config-server related tables in the udrConfigDB. So when upgrade is performed on the second UDR setup, then the following error occurs.

Figure 4-26 Config-Server Post Upgrade Hooks Error



- **OAuth2 Related Issues:**
  If you do not mention the OAuth secret name and namespace properly or if the public key certificate in secret is not in correct format, then the Ingress Gateway crashes.

Figure 4-27 Ingress Gateway Crashed



Other scenarios are:

- **The secret name in which public key certificate is stored is incorrect:** In this scenario, it is advisable to check the logs of a pod that states "cannot retrieve secret from api server".
- **The public key certificate stored in secret is not in proper format:** The public key format is `{nrfInstanceId}_RS256.crt (6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c_RS256.crt)`.
  If the public key is not stored in this format, then check the logs of pod that states "Malformed entry in NRF PublicKey Secret with key ecdsa.crt". Here, ecdsa.crt is the public key certificate in oauthsecret.

By using public key certificate in required format, you can resolve these issues. You need to correct the fields with the proper secret name and namespace.

4.3.3.4 Debugging PodDisruptionBudget Related Issues

A pod can have voluntary or involuntary disruptions at any given time. Voluntary disruptions are either initiated by the application owner or the cluster administrator. Examples of voluntary disruptions are deleting the deployment or a controller that manages the pod, updating a deployment pod template, or accidentally deleting a pod. Involuntary disruptions are unavoidable and can be caused due to any one or more of the following given examples.

- Disappearance of a node from the cluster due to cluster network partition
- Accidentally deleting a virtual machine instance
- Eviction of a pod when a node runs out of resources

To handle a voluntary disruption, you can set the **PodDisruptionBudget** value to determine the number of replicas of the application must be running at any given time. To configure the PodDisruptionBudget:

1. Run the following command to check the pods running on different nodes:
   ```
   kubectl get pods -o wide -n ocudr
   ```

Figure 4-28 Pods Running on Different Nodes

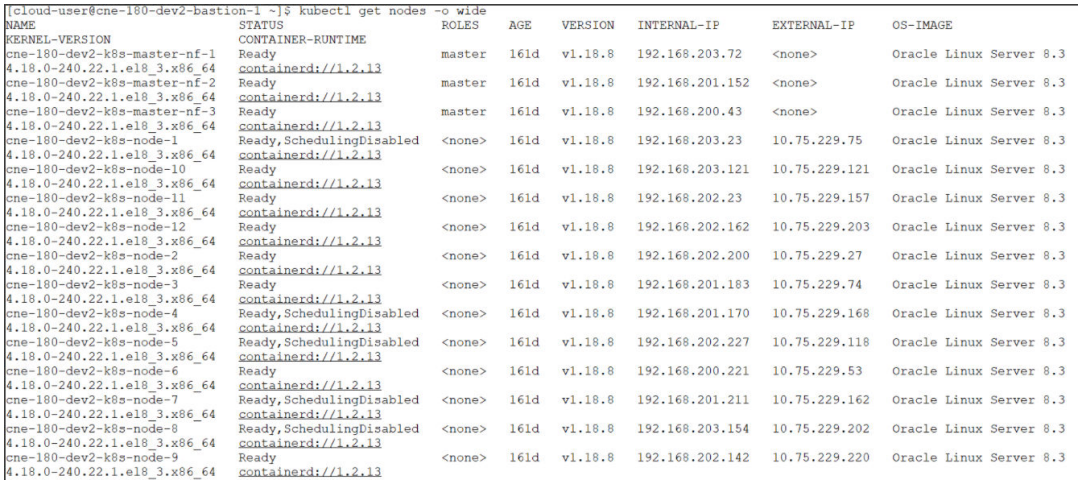
```
[cloud-user@cne-180-dev2-bastion-1 ~]$ kubectl get pods -o wide -n ocudr
NAME                                      READY   STATUS    RESTARTS   AGE   IP              NODE                      NOMINATED NODE   READINESS
GATES
ocudr-egressgateway-bcf776cd-gnhsb        1/1     Running   0          24m   10.233.94.62    cne-180-dev2-k8s-node-4   <none>           <none>
ocudr-ingressgateway-7bf9844bf-5hhq5      1/1     Running   0          24m   10.233.99.179   cne-180-dev2-k8s-node-7   <none>           <none>
ocudr-ingressgateway-7bf9844bf-nj6x5      1/1     Running   0          24m   10.233.97.75    cne-180-dev2-k8s-node-5   <none>           <none>
ocudr-nudr-config-7f64cddfc9-d9cgd        1/1     Running   0          24m   10.233.99.127   cne-180-dev2-k8s-node-7   <none>           <none>
ocudr-nudr-config-server-6f4cc7968b-tbrtt 1/1     Running   0          24m   10.233.102.52   cne-180-dev2-k8s-node-1   <none>           <none>
ocudr-nudr-diam-gateway-0                 1/1     Running   0          24m   10.233.97.78    cne-180-dev2-k8s-node-5   <none>           <none>
ocudr-nudr-diam-gateway-1                 1/1     Running   0          23m   10.233.93.227   cne-180-dev2-k8s-node-8   <none>           <none>
ocudr-nudr-diameterproxy-8d8cc5446-8p2cw  1/1     Running   0          24m   10.233.94.78    cne-180-dev2-k8s-node-4   <none>           <none>
ocudr-nudr-diameterproxy-8d8cc5446-gdvk9  1/1     Running   0          24m   10.233.93.136   cne-180-dev2-k8s-node-8   <none>           <none>
ocudr-nudr-drservice-564f66f9b5-6gtw8     1/1     Running   0          24m   10.233.102.34   cne-180-dev2-k8s-node-1   <none>           <none>
ocudr-nudr-drservice-564f66f9b5-fdg4g     1/1     Running   0          24m   10.233.93.125   cne-180-dev2-k8s-node-8   <none>           <none>
ocudr-nudr-notify-service-6cdd54bb7c-brhj4 1/1    Running   0          24m   10.233.99.73    cne-180-dev2-k8s-node-7   <none>           <none>
ocudr-nudr-notify-service-6cdd54bb7c-vbwgp 1/1    Running   0          24m   10.233.97.39    cne-180-dev2-k8s-node-5   <none>           <none>
[cloud-user@cne-180-dev2-bastion-1 ~]$
```

2. Run the following set of commands to unschedule the node:

   ```
   kubectl cordon cne-180-dev2-k8s-node-4
   kubectl cordon cne-180-dev2-k8s-node-7
   kubectl cordon cne-180-dev2-k8s-node-5
   kubectl cordon cne-180-dev2-k8s-node-8
   kubectl cordon cne-180-dev2-k8s-node-1
   ```

   After unscheduling the nodes, the state of the nodes changes to 'Ready,SchedulingDisabled' as follows:

Figure 4-29 After Nodes are Unscheduled



```
[cloud-user@cne-180-dev2-bastion-1 ~]$ kubectl get nodes -o wide
NAME                          STATUS                     ROLES    AGE   VERSION   INTERNAL-IP       EXTERNAL-IP     OS-IMAGE
KERNEL-VERSION                CONTAINER-RUNTIME
cne-180-dev2-k8s-master-nf-1  Ready                      master   161d  v1.18.8   192.168.203.72    <none>          Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-master-nf-2  Ready                      master   161d  v1.18.8   192.168.201.152   <none>          Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-master-nf-3  Ready                      master   161d  v1.18.8   192.168.200.43    <none>          Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-1       Ready,SchedulingDisabled   <none>   161d  v1.18.8   192.168.203.23    10.75.229.75    Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-10      Ready                      <none>   161d  v1.18.8   192.168.203.121   10.75.229.121   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-11      Ready                      <none>   161d  v1.18.8   192.168.202.23    10.75.229.157   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-12      Ready                      <none>   161d  v1.18.8   192.168.202.162   10.75.229.203   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-2       Ready                      <none>   161d  v1.18.8   192.168.202.200   10.75.229.27    Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-3       Ready                      <none>   161d  v1.18.8   192.168.201.183   10.75.229.74    Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-4       Ready,SchedulingDisabled   <none>   161d  v1.18.8   192.168.201.170   10.75.229.168   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-5       Ready,SchedulingDisabled   <none>   161d  v1.18.8   192.168.202.227   10.75.229.118   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-6       Ready                      <none>   161d  v1.18.8   192.168.200.221   10.75.229.53    Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-7       Ready,SchedulingDisabled   <none>   161d  v1.18.8   192.168.201.211   10.75.229.162   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-8       Ready,SchedulingDisabled   <none>   161d  v1.18.8   192.168.203.154   10.75.229.202   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
cne-180-dev2-k8s-node-9       Ready                      <none>   161d  v1.18.8   192.168.202.142   10.75.229.220   Oracle Linux Server 8.3
4.18.0-240.22.1.el8_3.x86_64  containerd://1.2.13
```

3. Run the following set of commands to drain the nodes:

```
kubectl drain cne-180-dev2-k8s-node-1 --ignore-daemonsets --delete-local-
data
kubectl drain cne-180-dev2-k8s-node-8 --ignore-daemonsets --delete-local-
data
kubectl drain cne-180-dev2-k8s-node-5 --ignore-daemonsets --delete-local-
data
kubectl drain cne-180-dev2-k8s-node-4 --ignore-daemonsets --delete-local-
data
kubectl drain cne-180-dev2-k8s-node-7 --ignore-daemonsets --delete-local-
data
```

4. If you are required to drain the nodes or evict the pods, then ensure the minimum number of pods are in ready state to serve the application request. To configure the minimum number of pods value, set the **minAvailable** parameter in the Helm charts for individual microservices. This ensures the availability of a minimum number of pods and they are not evicted. You can check logs while draining the nodes as follows:

Figure 4-30 Logs When Trying to Evict Pod

```
error when evicting pod "ocudr-nudr-notify-service-6cdd54bb7c-brhj4" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption
budget.
error when evicting pod "ocudr-nudr-config-7f64cddfc9-d9cgd" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
error when evicting pod "ocudr-nudr-diam-gateway-0" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
error when evicting pod "ocudr-ingressgateway-7bf9844bf-nj6x5" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
error when evicting pod "ocudr-nudr-diameterproxy-8d8cc5446-8p2cw" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption
budget.
error when evicting pod "ocudr-nudr-diameterproxy-8d8cc5446-gdvk9" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption
budget.
error when evicting pod "ocudr-nudr-drservice-564f66f9b5-fdg4g" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption
budget.
error when evicting pod "ocudr-nudr-drservice-564f66f9b5-6gtw8" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption
budget.
error when evicting pod "ocudr-nudr-notify-service-6cdd54bb7c-vbwgp" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption
budget.
error when evicting pod "ocudr-ingressgateway-7bf9844bf-5hhq5" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
error when evicting pod "ocudr-nudr-config-server-6f4cc7968b-tbrtt" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption
budget.
error when evicting pod "ocudr-nudr-diam-gateway-1" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
error when evicting pod "ocudr-egressgateway-bcf776cd-gnhsb" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
```

4.3.3.5 Debugging Pod Eviction Issues

During heavy traffic, there can be a situation where any UDR or Provisioning Gateway pod can run into evicted state. To handle the eviction issues, increase the Ephemeral storage allocation of pods under global section. Update the **containerLogStorage** configuration under **global.ephemeralStorage.limits** section to '5000'.

Figure 4-31 Configuring Container Log Storage

```
# Ephemeral Storage for all Containers, Units in MB, Keep Reference Definition Unchanged
ephemeralStorage:
  requests:
    containersLogStorage: &containersLogStorageRequestsRef 50
    containersCrictlStorage: &containersCrictlStorageRequestsRef 2
  limits:
    containersLogStorage: &containersLogStorageLimitsRef 1000
    containersCrictlStorage: &containersCrictlStorageLimitsRef 2
```

After making the above changes, perform helm upgrade. If the Ingress Gateway or Egress Gateway pods are running into Evicted state, then update the **ephemeralStorageLimit** configuration to '5120' and perform a helm upgrade.

Figure 4-32 Ingress Gateway or Egress Gateway - Evicted State

```
# Ephemeral storage configuration for log storage
logStorage: *containersLogStorageRequestsRef
# Ephemeral storage configuration for crictl storage
crictlStorage: *containersCrictlStorageRequestsRef
# Ephemeral Storage Limit Configuration
ephemeralStorageLimit: 1024
```

4.3.3.6 Debugging Taints or Tolerations Misconfigurations

The following points should be considered when the Node Selector and Taints or Toleration feature is used on UDR or Provisioning Gateway:

- If any of the pods are going to the Pending state, ensure that the node selector used is pointing to the correct slave node name, and it has enough space to place the pod.

  Figure 4-33 Global Configuration

```
nodeSelection: DISABLED
# Toleration setting for all microservices. Set tolerationsSetting to ENABLED
tolerations: []
# Below v1 setting is used in all CNEs.
# Do not set nodeSelection to ENABLED to use this flag
nodeSelector:
  nodeKey: ''
  nodeValue: ''
```

- Use the following configuration to configure toleration settings for the tainted nodes.Update the Global section configurations for the settings to be applicable for all the services.

  Figure 4-34 Global Configuration

```
tolerationsSetting: DISABLED
# Flag to enable global level nodeSelection setting.
# If set to ENABLED and local nodeSelection setting at each micro service
# level is set to USE_GLOBAL_VALUE the same will be used in the microservice
nodeSelection: DISABLED
# Toleration setting for all microservices. Set tolerationsSetting to ENABLED
tolerations: []
# Below v1 setting is used in all CNEs.
# Do not set nodeSelection to ENABLED to use this flag
```

- For Node Selector and Tolerations, configuration at the microservice level takes priority over configuration at the global level.

Figure 4-35 Toleration and Node Selector

```
# Toleration and Node Selector
tolerationsSetting: USE_GLOBAL_VALUE
nodeSelection: USE_GLOBAL_VALUE
tolerations: []
helmBasedConfigurationNodeSelectorApiVersion: "v1"
nodeSelector:
  nodeKey: ''
  nodeValue: ''
# Uncomment below configuration if v2 is used with NodeSelector
#nodeSelector: {}
```

4.3.3.7 Debugging UDR Registration with NRF Failure

UDR registration with NRF may fail due to various reasons. Some of the possible scenarios are as follows:

- Confirm whether registration was successful from the nrf-client-service pod.
- Check the ocudr-nudr-nrf-client-nfmanagement logs. If the log has **"UDR is Deregistration"** then:
  - Check if all the services mentioned under allorudr/slf (depending on UDR mode) in the values.yaml file has same spelling as that of service name and are enabled.
  - Once all services are up, UDR must register with NRF.
- If you see a log for **SERVICE_UNAVAILABLE(503)**, check if the primary and secondary NRF configurations (primaryNrfApiRoot/secondaryNrfApiRoot) are correct and they are UP and Running.

4.3.3.8 Debugging User Agent Header Related Issues

If there are issues related to user agent header, then perform the following steps:

- Under Ingressgateway section, enable the **user-agent** flag to true.
- If there are issues in the consumer NFTypes validations for NF's, then check the NF types in the configurations present under ingressgateway section.

Figure 4-36 Enabling User Agent Header

```
#User-Agent header validator configuration
#Mode of configuration. Can be either HELM or REST
userAgentHeaderValidationConfigMode: HELM
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 perf
  # 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"
```

- If the `userAgentHeaderValidationConfgMode` set to REST, the custom-values.yaml configurations are ignored. The configuration is loaded based on `userAgentHeaderValidationConfgMode` is set.

> **Note**
>
> From UDR 24.1.0 release onwards this feature is supported using REST API mode. If the feature does not work, make sure the feature is enabled and configured post UDR is upgraded, see Postupgrade Tasks section in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*

4.3.3.9 Debugging LCI and OCI Header Related Issues

If there are issues related to the LCI and OCI Header feature, perform the following:

Under Ingressgateway-sig section, set the `lciHeaderConfig.enabled` and `ociHeaderConfig.enabled` parameters as `true`, respectively.

> **Note**
>
> - Configured the names of the headers the same as in the default configuration.
> - Ensure to wait upto the validity period to report LCI and OCI Header in the next response.
> - For OCI, check the overloadconfigrange and reduction metrics. Based on which, OCI is reported.

4.3.3.10 Debugging Conflict Resolution Feature

Perform the following steps if the conflict resolution feature does not work:

- If the exception tables or UPDATE_TIME column are missing from the UDR subscriber database, perform the following steps:
  - Ensure to run the SQL command from the SQL files on the database ndbappsql node.

    > **Note**
    >
    > Following SQL files are available in Custom_Templates 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*.

  - after setting the `global.dbConflictResolutionEnabled` parameter to true in ocudr_custom_values.yaml file, if the UPDATE_TIME column is updated as 0, then run

the REST APIs to fetch the global configurations and check the `global.dbConflictResolutionEnabled` is set to true. If the parameter is not set to true, perform a PUT operation for global configuration update to set the parameter to true.

- If nudr_dbcr_auditor service is not enabled, then make sure to enable the `global.dbConflictResolutionEnabled` parameter and perform an Helm upgrade.
- If nudr_dbcr_auditor service is not clearing exception tables or fixing data conflict, then make sure that the database replication is running.
- If nudr_dbcr_auditor service is not clearing exceptions on IDXTODATA$EX exception tables then you must check if have the error log "Communication failed to Mate site during audit". If you get this error make sure to check if the following configuration on the custom values file is configured correctly.

```
# 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'
```

4.3.3.11 Debugging UDR Error Responses Using App Error Code

From UDR release 24.1.0 onwards additional information are provided in the `ProblemDetails.detail` parameter as part of the error responses feature. This is applicable for all UDR mode deployments which includes signaling and provisioning error responses from UDR.

> **Note**
>
> The responses from nudr-config service for the REST APIs configuration remains same.

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`

Table 4-2 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`

Table 4-2 (Cont.) Parameters of the Details Field of the Payload

Parameter Name	Description
App-Error ID	Indicates the microservice ID and the error IDo<nftype>-<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

4.3.3.12 Debugging Provisioning Logs Related Issues

If there are issues related to provisioning log, then perform the following steps:

- You can enable or disable the provisioning log feature by setting the `provLogsEnabled` parameter flag to true using REST APIs, CNC Console, or by changing the values in custom.yaml file. By default provision logging feature is disabled.
- You can set the provisioning API names that are supported for provision logging by changing the `provLogsApiNames` configuration field to the required value. The default value is nudr-dr-prov. The accepted values are as follows:
  - nudr-dr-prov
  - nudr-group-id-map-prov
  - slf-group-prov
  - n5g-eir-prov
- If `provLogsEnabled` flag is set to true, then it is recommended to change the values of logStorage to 4000 MB (apporx. equal to 4GB) for nudr-dr-prov pods to store provision logging files. If the values are not updated, then nudr-dr-prov pods will crash when ephemeral storage is full.

```
# Resource specification for nudr-drservice container
# If provLogsEnabled is true, then change the values of logStorage to 4000 for storing provLogs in nudr-dr-prov pods.
resources:
  limits:
    cpu: 2
    memory: 2Gi
    logStorage: *containersLogStorageLimitsRef
    crictlStorage: *containersCrictlStorageLimitsRef
  requests:
    cpu: 2
    memory: 2Gi
    logStorage: *containersLogStorageRequestsRef
    crictlStorage: *containersCrictlStorageRequestsRef
```

4.3.4 Debugging Upgrade or Rollback Failure

When Unified Data Repository (UDR) upgrade or rollback fails, perform the following steps:

1. Run the following command to check the pre or post upgrade or rollback hook logs:

   ```
   kubectl logs <pod name> -n <namespace>
   ```

2. After detecting the cause of failure, do the following:
   - For upgrade failure:
     - If the cause of upgrade failure is database or network connectivity issue, then resolve the issue and rerun the upgrade command.
     - If the cause of failure is not related to database or network connectivity issue and is observed during the preupgrade phase, then do not perform rollback because UDR deployment remains in the source or previous release.
     - If the upgrade failure occurs during the post upgrade phase. For example, post upgrade hook failure due to target release pod not moving to ready state, then perform a rollback.
   - For rollback failure: If the cause of rollback failure is database or network connectivity issue, then resolve the issue and rerun the rollback command.
3. If the issue persists, contact My Oracle Support.

4.4 Service Related Issues

This section describes the most common service related issues and their resolution steps.

4.4.1 Resolving Microservices related Issues through Metrics and ConfigDB

This section describes how to troubleshoot issues related to UDR microservices using metrics.

nudr-drservice

If requests for nudr-drservice fail, then try to find the root cause from metrics using following guidelines:

- If the count of measurement “udr_schema_operations_failure_total” is increasing, check the content of the incoming request and make sure that the incoming json data blob is proper and as per the specification.
- If “udr_db_operations_failure_total” measurements are increasing,

- Make sure that connectivity is proper between microservices and MySQL DB nodes.
- Make sure that you are not trying to insert duplicate keys.
- Make sure that DB nodes have enough resources available.

nudr-dr-provservice

If requests for nudr-dr-provservice fails, then try to find the root cause from metrics using following guidelines:

- If the count “udr_schema_operations_failure_total” measurement is increasing, check the content of incoming request and ensure the incoming JSON data blob is proper and as per the specification.
- If “udr_db_operations_failure_total” measurements are increasing, then ensure:
  - there is connectivity between microservices and MySQL DB nodes.
  - you are not trying to insert duplicate keys.
  - database nodes have enough resources available.

nudr-nrf-nfmanagement

If requests for nudr-nrf-nfmanagement fail, then try to find the root cause from metrics using following guidelines:

- Check for current health status of NRF using the `nrfclient_nrf_operative_status` metric. If it is 0, it is UNHEALTHY or UNAVAILABLE.
- Check for current NF status using the `nrfclient_nrf_operative_status` metric, and NF status with NRF with `nrfclient_nf_status_with_nrf` metric.
- If NF status is 0, then check the `appinfo_service_running` metric for various services configured in the app-info section depending on the UDR mode.

nudr-nrf-client-service

If requests for nudr-nrf-client-service fail, then try to find the root cause from metrics using following guidelines:

- Check the current health status of NRF using "nrfclient_nrf_operative_status" metric. If it is 0, then it is UNHEALTHY or UNAVAILABLE.
- Check the current network function status using "nrfclient_nrf_operative_status" metric and network function status with NRF using "nrfclient_nf_status_with_nrf" metric.
- If network function status is 0, check "appinfo_service_running" metric for various services configured in the app info section depending on UDR mode.

ocudr-nudr-notify-service

If requests for ocudr-nudr-notify-service fail, then try to find the root cause from metrics using following guidelines:

- Measurements like “nudr_notif_notifications_ack_2xx_total”, “nudr_notif_notifications_ack_4xx_total”, “nudr_notif_notifications_ack_5xx_total” gives information about the response code returned in the notification response.
- If count of “nudr_notif_notifications_send_fail_total” measurement is increasing, make sure that notification server mentioned in NOTIFICATION_URI during subscription request, which is expected to receive the notifications, is up and running.

- The default retry count for failed notifications is two and this is configurable from the `retrycount` parameter in the custom values yaml file. Perform the following steps if alerts is raised for exceeding notifications table limit threshold:
  - Log in to mysql database terminal to check the number of records on the NOTIFICATIONS table under UDR subscriber database (select count(*) from NOTIFICATIONS).
  - Perform the following steps if the notification records count is consistently above 50k:
    * Check if there are more failures on the notification sent from notify service using `nudr_notif_notifications_ack_4xx_total` and `nudr_notif_notifications_ack_5xx_total` metrics.
    * Check the reason for the failure and resolve the failure.
    * If the failure is temporary and cannot be avoided then use the notifications configuration REST API or CNC Console to reduce the `retrycount` to 0 or 1. This will make sure that the table size does not increase faster.

ocudr-nudr-config

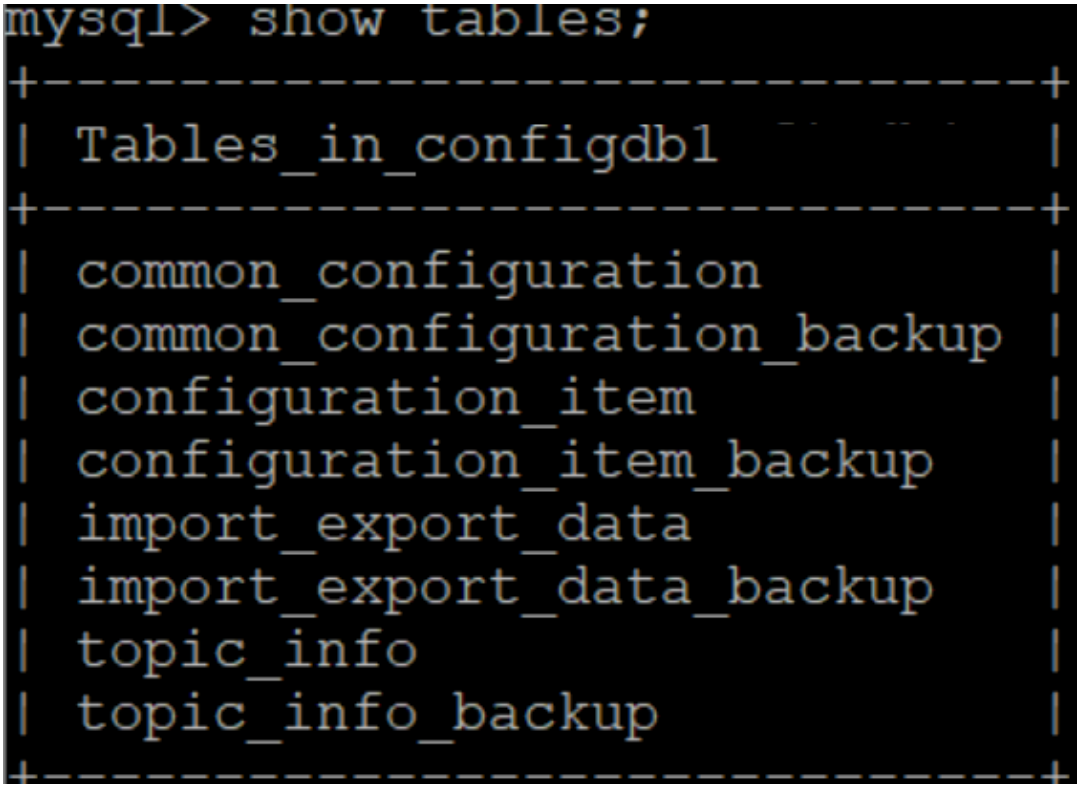
If requests for ocudr-nudr-config fail, try to find the root cause from metrics using following guidelines:

- Measurements like “nudr_config_total_requests_total{Method='GET'}”, “nudr_config_total_requests_total{Method='POST'}”, “nudr_config_total_requests_total{Method='PUT'}” gives information about the total request pegged for the method GET, POST, and PUT respectively.
- If count of measurement “nudr_config_total_responses_total{Method='GET/POST/PUT',StatusCode="400/404/405/500"}” is increasing, it means the requests are not being processed and results in failures.

If requests for ocudr-nudr-config fail, try to find the root cause from configdb using following guidelines:

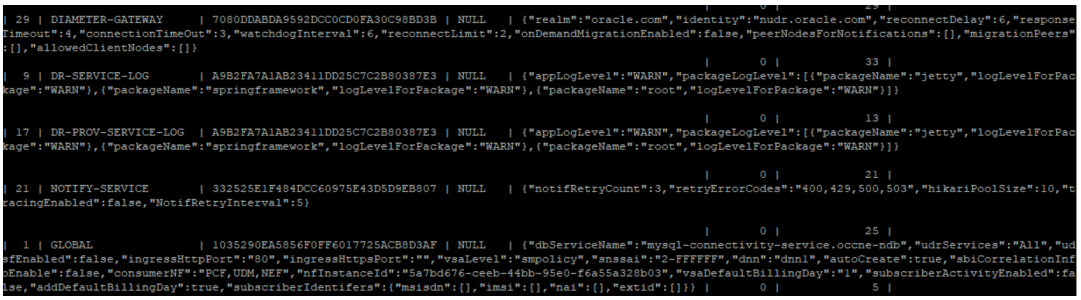
- If you get a **BAD REQUEST** for GET API, then make sure all the tables shown below is present in configdb table.

Figure 4-37 Configdb Table



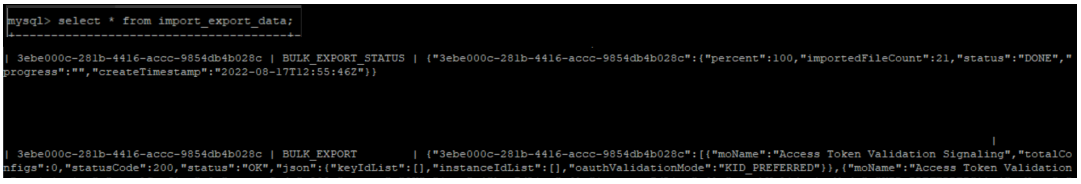
- If all the table are present and you are getting a **BAD REQUEST** for GET API, then you must verify the configuration item table shown below.

Figure 4-38 Configuration Item Table



- If you get a **BAD REQUEST** and **NOT FOUND** for Import and Export API, then you must verify the import and export data table shown below.

Figure 4-39 Import and Export Data



ocudr-nudr-bulk-import

Following are some of the known errors that you can address if encountered.

- If the bulk-import logs show "dr-service is down. Job cannot be executed", then check whether dr-service and Ingress Gateway are in the running state.
- If the count of **nudr_bulk_import_csvfile_records_read_total(Method="DELETE/PUT/POST", Status="Failure")** metric is increasing, then it means the CSV file records are not valid. This can be resolved by providing correct keyType, KeyValue, operationType, nfType, and jsonPayload.
- If the count of **nudr_bulk_import_records_processed_total(Method = "POST/PUT/DELETE", StatusCode="201/204", Status="Success")** is increasing, then it means the records are being processed by UDR correctly.
- To find the number of request processed successfully for PCF, measure the count of **Nudr_bulk_import_PCF_total{StatusCode="204/201", Status="Success"}** metric.

For information about bulk import metrics, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.

ocudr-nudr-xmltocsv

After copying the ixml file using kubectl cp command, log into xmltocsv container and run the following command to check whether the file is copied or not:

```
> kubectl exec -it <pod name> -c nudr-xmltocsv -n <namespace> bash > cd /home/udruser/xml
```

If the count of measurement of the **nudr_xmltocsv_xmlfile_records_read_total(Status="Failure")** metric is increasing, then it shows the records in the ixml file are not valid. You need to ensure that correct ixml file is provided.

If the measurement count of the **nudr_xmltocsv_records_processed_total{Method = "POST/PUT/DELETE/PATCH", Status="Success"}** metric is increasing, then it denotes that the records are processed successfully.

For information about xmltocsv metrics, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.

ocudr-nudr-diameterproxy

If diameterproxy restarts, then make sure the database configurations are correct. For information about ocudr-nudr-diameterproxy metrics, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.

diam-gateway

If the Diameter Gateway sends a CEA message with DIAMETER_UNKNOWN_PEER metric, then it means the client peer configuration is not done correctly. Configure the **allowedClientNodes** section of Diameter Gateway service using REST API.

If the Diameter Gateway sends a CEA message success and other SH message response with DIAMETER_UNABLE_TO_COMPLY/DIAMETER_MISSING_AVP metric, then the problem may lie in the requested Sh message.

If the Diameter Gateway error logs show errors like connection refused with some IP and port, then it means a specified peer node configured is not able to accept the CER request from the Diameter Gateway and Diameter Gateway retries to connect with that peer.

If you are getting **DIAMETER_UNABLE_TO_DELIVERY** error message, then it means diameterproxy is turned off or not running. If the Diameter Gateway goes to crashloop back off state, then it means that incorrect peer node is configured.

Use metric ocudr_diam_conn_network to verify the active connection in the peer nodes.

For information about diam-gateway metrics, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.

nudr-migration

If a pod is in the **pending** state, it means resources are not present in the CNE and if a pod is in the **ImagePullBackoff** state, it means the image is not able to fetch from repository. Run the following command to check details:

```
kubectl describe pod <pod-name> -n <namespace>
```

If the pod is in the **running** state and data migration has not happened, then:

- check the logs and search for ERROR in logs
- Either the source UDR or target UDR is down. Verify logs.

If you are not able to connect to 4G UDR, then:

- Check logs for **DIAMETER_UNABLE_TO_COMPLY** in CER/CEA messages.
- Check whether **UDR/UDA** messages are received from 4G UDR.
- Check whether **K8S_HOST_IP** port is same as an external IP address of Kubernetes node that you gave in affinity. If they are different, then you get **DIAMETER_UNABLE_TO_COMPLY** in CEA response.

For information about nudr-migration metrics, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.

overload-manager

To troubleshoot errors related to overload-manager, consider the following points:

- In the global section, if the **overloadmanager** flag is disabled, then the overload manager REST APIs of Ingress Gateway and perf-info microservice do not load.
- If the overload manager data is not present in the common_configuration table, then ensure the **overloadmanger** flag is enabled at the global level.
- **svcName** configured at ocpolicymapping API should be taken from routesConfig section. If the svcName configured in policymapping is different from svcName configured in routesConfig, then overload manager does not trigger.
- To check specific load level of metric, check the perf-info logs. The perf-info logs contain load level of each metric.
- If the alerts are not raised for overload manager, then ensure the alerts are properly loaded and are not loaded from Prometheus.

On-demand migration Range Support

To troubleshoot errors related to on-demand migration range support, consider the following points:

- By default, on-demand migration works for all key type and key values, if there is no change in the configurations. Check the REST configuration of global section for key type and key range.
- If on-demand migration does not trigger after key type and key range is set through global configuration API, perform the following step:
  - Check if the valid key type and key range that is mentioned in the configuration API contains the same key type and key range that is used for the test. Valid keys are Mobile Station Integrated Services Digital Network (MSISDN) or International Mobile Subscriber Identity (IMSI).
- If the on-demand migration range support feature is not used, you can set the default key type and key range from the global configuration API as below:

```
"keyType": "msisdn",
"keyRange": "000000-000000"
```

For information about on-demand migration metrics, see *Oracle Communications Cloud Native Core, Unified Data Repository Users Guide*.

4.4.2 Debugging Errors from Egress Gateway

If the traffic is not routed through Egress Gateway, then check the following:

- Check whether `global.egress` is enabled.
- Check whether Egress pod is running from kubectl. To check, run the following command: `kubectl get pods -n <Release.name>`
- To enable the outgoing traffic using HTTPS, set the **enableOutgoingHttps** parameter as 'true'.
- Create unique certificates and keys for all Egress and respective Ingress NF's. It is the same as Ingress debugging.

Debugging Errors When SCP Integration is Enabled

UDR Egress Gateway route configurations are performed to route all the notifications through SCP, and the NRF traffic is sent directly to the NRF host. If the routing does not work, then configure the routes as follows:

Figure 4-40 Routes Config

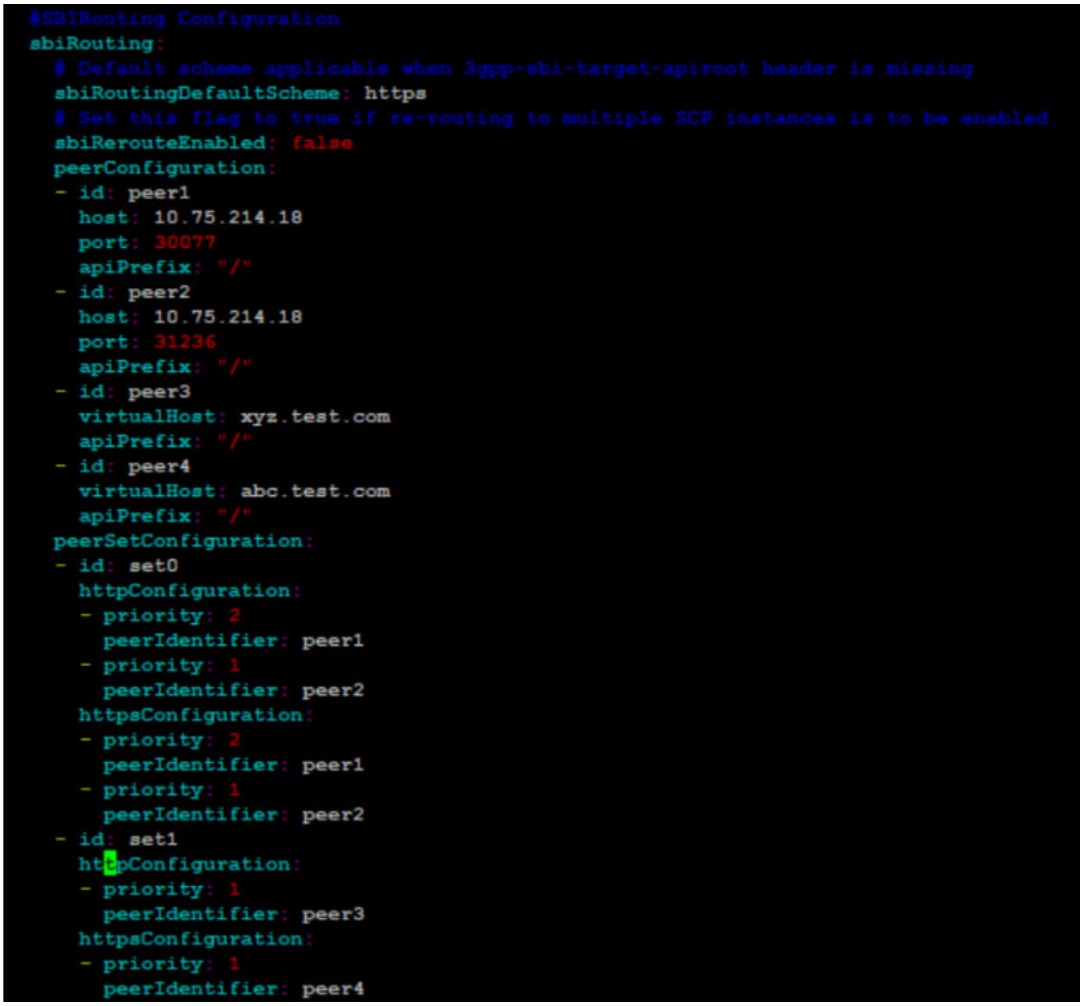
```
routesConfig:
# Route for NRF traffic
- id: nrf_nfm_direct
  uri: egress://dummy.dontchange
  path: /nnrf-nfm/**
  order: 1
# Route for Notification traffic
- id: notifications_direct
  uri: http://dummy.dontchange
  path: /**
  order: 2
  metadata:
    httpsTargetOnly: false
    httpRuriOnly: false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

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

> **Note**
>
> The above configuration is present as part of default values.

If you want to send notifications through SCP, configure Egress Gateway as shown in the following image. If **setId** 0 is used, configure both **httpConfigs** and **httpsConfigs** as shown in the image. For setId having static host configuration for httpsConfigs (even if its not used), it is mandatory to configure this parameter using dummy values as shown in the image. If it is not configured, then the Egress Gateway log shows NullPointerException.

Figure 4-41 Sending Notification Through SCP



If it uses setId 1 or 2, enable Alternate Route service and configure proper host details for Egress Gateway to communicate with alternate route service. If configurations are not done as expected, then it gives 425 error, which is the default error configured for virtual FQDN lookup failure. If you see 503 or other 4xx errors, then it is because the actual endpoint or SCP is not reachable.

Figure 4-42 Using setId 1 or 2

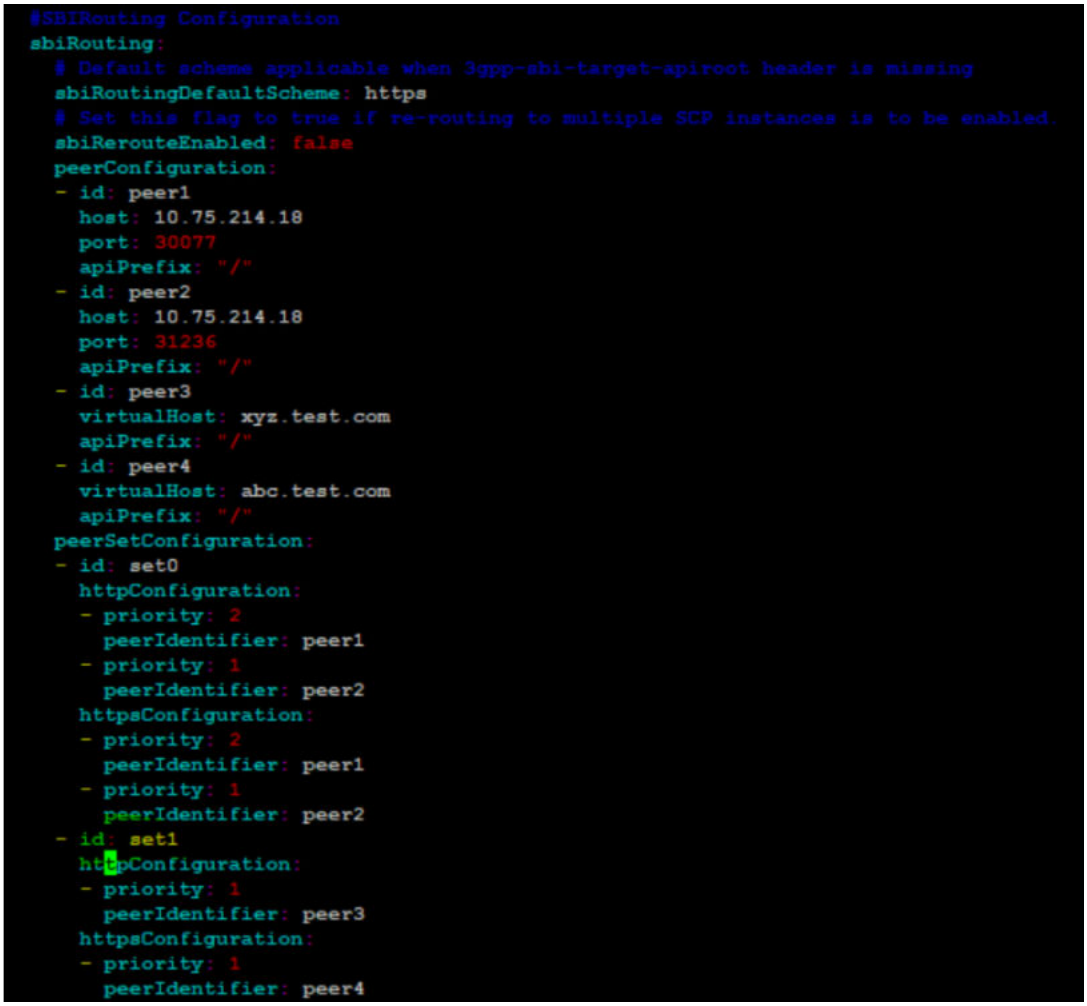


Figure 4-43 Using setId

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

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

Figure 4-44 Using setId 1 or 2 (cont..)

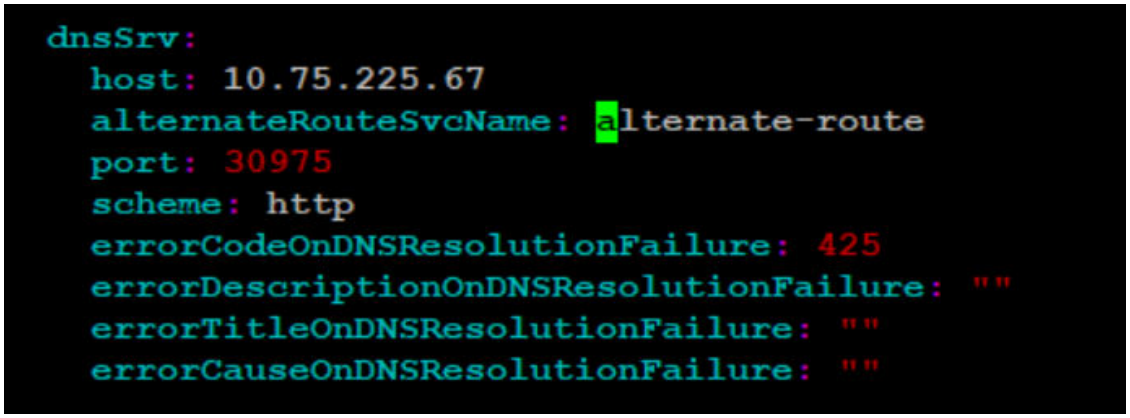
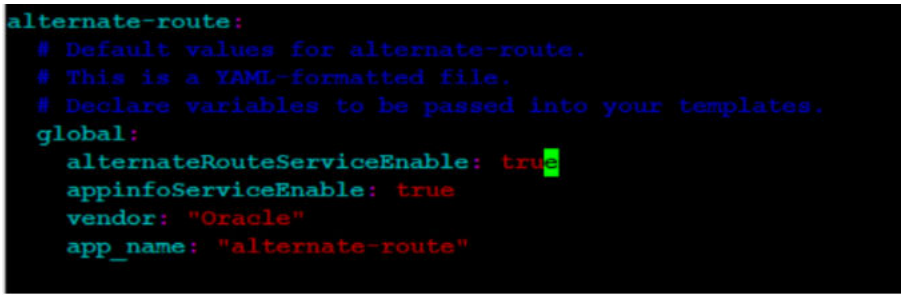
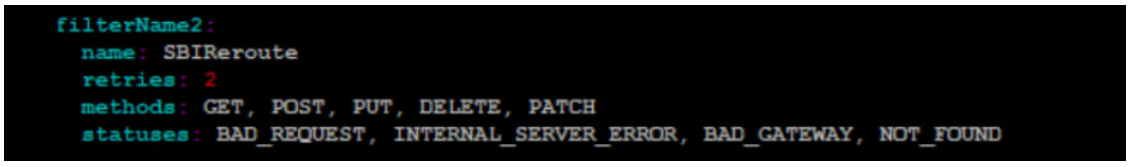


Figure 4-45 Using setId 1 or 2 (cont..)



Retry to multiple SCPs in case of failure depends on the failure code and operation performed. If it is not in the configured list, then it does not attempt a retry. The number of retries depends on retries configuration as follows:

Figure 4-46 SCP Retry



Also, ensure that **scpRerouteEnabled** is set to true.

Figure 4-47 scpRerouteEnabled set to true

If DNS resolution from core-dns service does not happen, check whether the following configuration is enabled on alternate-route service.

Figure 4-48 DNS Srv Configuration

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

4.4.3 Debugging Errors from Ingress Gateway

The possible errors that you may encounter from Ingress Gateway are:

- **Check for 404 Error:** If the request fails with 404 status code with the following ProblemDetails, then there may be issues with the routeConfig on the ingressgateway custom values file.

  ```
  {"title":"404 NOT_FOUND","status":404,"detail":"udr001.oracle.com:
  ingressgateway: NOT_FOUND: OUDR-IGWSIG-E183"}
  ```

  You must check the custom values.yaml file for the essential route configurations. If the essential route configurations are not present you must add the route configurations.

- **Check for 503 Error:** If the request fails with 503 status code with "SERVICE_UNAVAILABLE" in Problem Details, then it means that the nudr-drservice pod is not reachable due to some reason.

  ```
  {"title":"Service Unavailable","status":503,"detail":"udr001.oracle.com:
  ingressgateway: Service Unavailable: OUDR-IGWSIG-
  E003","cause":"Encountered unknown host exception at IGW"}
  ```

  You can confirm the same in the errors/exception logs of the ocudr-ingressgateway pod. Check for ocudr-nudr-drservice pod status and fix the issue.

4.4.4 Debugging Errors from nudr-config

The Cloud Native Core (CNC) Console GUI uses the debugging errors received from nudr_config to update or view the configuration items. The debugging error details from nudr-config are as follows:

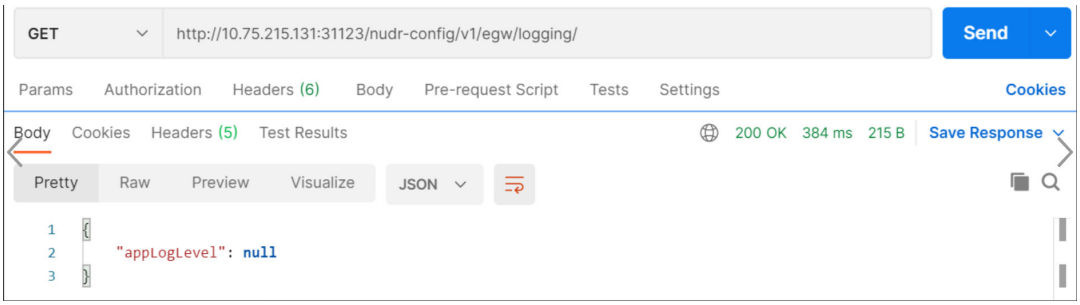
- **Check for 400 Error:** If the following request fails with 400 status code with "404 Not Found", it indicates that the logging level information is not present in the database or the microservice is not enabled.

  Figure 4-49 Checking for 400 Error

```
[apigateway@ocudr-egressgateway-bfb97c974-p6kfj oracle]$ curl -v -X GET http://ocudr-nudr-config:5001/nudr-config/v1/nudr-ondemand-migration/logging
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 10.233.48.14...
* TCP_NODELAY set
* Connected to ocudr-nudr-config (10.233.48.14) port 5001 (#0)
> GET /nudr-config/v1/nudr-ondemand-migration/logging HTTP/1.1
> Host: ocudr-nudr-config:5001
> User-Agent: curl/7.61.1
> Accept: */*
>
< HTTP/1.1 404 Not Found
< Connection: keep-alive
< ocLogId: 1725436292420_8914_ocudr-nudr-config-7bf89948b-7f9lx
< Transfer-Encoding: chunked
< Content-Type: application/problem+json
< Date: Wed, 04 Sep 2024 07:51:32 GMT
<
* Connection #0 to host ocudr-nudr-config left intact
{"title":"Not Found","status":404,"detail":"nudr-config: Data field does not exist: OUDR-DRC-CFG-E565","cause":"DATA_NOT_FOUND","invalidParams":[]}[
```

If **common_config_hook** is unable to create configuration item for the common services like ingress-gateway, egress-gateway, or alternate-route, then the GET request for the logging gives the following response:

Figure 4-50 Response of Get Request for Logging



4.4.5 Debugging Notification Issues

If UDR does not generate any notification, check the notify service port configuration in the values.yaml file. These ports must be same as the ports on which notify service is running.

```
nudr-drservice:
...
...
...
    notify:
        port:
            http: 5001
            https: 5002
```

5 Alert Configuration

This section describes how to configure alert rules for the UDR. It provides guidance on setting up measurement-based alert rules, where the alerting system evaluates metrics reported by UDR microservices against specified rule conditions to generate alerts as needed. UDR alert rules are configured based on metrics reported by UDR components. The alerting workflow monitors these metrics and issues notifications when the defined conditions are met. For more information about configuring UDR alerts in Prometheus, see the “Alert Configuration” section in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

5.1 Alert Details

This section describes alerts in detail.

> **Note**
>
> Max Ingress requests/sec in consideration is 1000/second.

Table 5-1 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.
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 5-2 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
NudrNRFClientServiceDown	NudrNRFClientServiceDown
NudrConfigServiceDown	NudrConfigServiceDown
NudrDiameterProxyServiceDown	NudrDiameterProxyServiceDown
NudrOnDemandMigrationServiceDown	NA
OcudrIngressGatewayServiceDown	OceirIngressGatewayServiceDown
OcudrEgressGatewayServiceDown	OceirEgressGatewayServiceDown
OcudrDbServiceDown	OceirDbServiceDown
OcudrXFCCValidationFailureAbove10Percent	OceirXFCCValidationFailureAbove10Percent
OcudrXFCCValidationFailureAbove20Percent	OceirXFCCValidationFailureAbove20Percent
OcudrXFCCValidationFailureAbove50Percent	OceirXFCCValidationFailureAbove50Percent
DRServiceOverload60Percent	DRServiceOverload60Percent
DRServiceOverload75Percent	DRServiceOverload75Percent
DRServiceOverload80Percent	DRServiceOverload80Percent
DRServiceOverload90Percent	DRServiceOverload90Percent

Table 5-2 (Cont.) Alert names for UDR/SLF and EIR

UDR/SLF	EIR
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
PVCFullForUDRExport	NA
UDRExportFailed	NA
IngressgatewayPodProtectionDocState	IngressgatewayPodProtectionDocState
IngressgatewayPodProtectionCongestedState	IngressgatewayPodProtectionCongestedState
RetryNotificationRecordsMaxLimitExceeded	RetryNotificationRecordsMaxLimitExceeded
UserAgentHeaderNotFoundMorethan10PercentRequest	NA
EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold	EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold
EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold	EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold

Table 5-2 (Cont.) Alert names for UDR/SLF and EIR

UDR/SLF	EIR
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 5-2.

5.1.1 System Level Alerts

This section lists the system level alerts.

5.1.1.1 OcudrSubscriberNotFoundAbove1Percent

Table 5-3 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
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.

5.1.1.2 OcudrSubscriberNotFoundAbove10Percent

Table 5-4 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.

5.1.1.3 OcudrSubscriberNotFoundAbove25Percent

Table 5-5 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
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:** udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

5.1.1.4 OcudrSubscriberNotFoundAbove50Percent

Table 5-6 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	udr_subscriber_not_found_total
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:** udr_rest_failure_response_total 2. If guidance required, Contact My Oracle Support.

5.1.1.5 OcudrPodsRestart

Table 5-7 OcudrPodsRestart

Field	Details
Description	Pod {{$labels.pod}} has restarted.
Summary	namespace: {{$labels.namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : 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	kube_pod_container_status_restarts_total

Table 5-7 (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 like connectivity, kubernetes secrets and so on. 2. Check orchestration logs for liveness or readiness probe failures using the following commands. *kubectl get po -n <namespace>* Note the full name of the pod that is not running and use it in the following command. *kubectl describe pod <desired full pod name> -n <namespace>* 3. Check the DB status. For more information, see *Oracle Communications Cloud Native Core DBTier User Guide*. 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 *Oracle Communications Cloud Native Core Network Function Data Collector User's Guide*.

5.1.1.6 NudrServiceDown

Table 5-8 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-drservice

Table 5-8 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.7 NudrProvServiceDown

Table 5-9 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 5-9 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.8 NudrNotifyServiceServiceDown

Table 5-10 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 5-10 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.9 NudrNRFClientServiceDown

Table 5-11 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="nudr-nrf-client-service"

Table 5-11 (Cont.) NudrNRFClientServiceDown

Field	Details
Recommended Actions	The alert is cleared when the NRFClientService service is available. **Steps:** 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures using the following commands. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.10 NudrConfigServiceDown

Table 5-12 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 5-12 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.11 NudrDiameterProxyServiceDown

Table 5-13 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 5-13 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.12 NudrOnDemandMigrationServiceDown

Table 5-14 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 5-14 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.13 OcudrIngressGatewayServiceDown

Table 5-15 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 5-15 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.14 OcudrEgressGatewayServiceDown

Table 5-16 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 5-16 (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. *kubectl get po -n <namespace>* Note the full name of the pod that is not running. It must be used in the following command *kubectl describe pod <specific desired full pod name> -n <namespace>* 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.

5.1.1.15 OcudrDbServiceDown

Table 5-17 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.

5.1.1.16 OcudrIngressGatewayProvServiceDown

Table 5-18 OcudrIngressGatewayProvServiceDown

Field	Details
Description	OCUDR Ingress-Gateway service {{$labels.app_kubernetes_io_name}} is down

Table 5-18 (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:
	`kubectl get po -n <namespace>`
	Note the full name of the pod that is not running. It must be used in the following command:
	`kubectl describe pod <specific desired full pod name> -n <namespace>`
	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.

5.1.2 Application Level Alerts

This section lists the application level alerts.

5.1.2.1 OcudrTrafficRateAboveMajorThreshold

Table 5-19 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 5-19 (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.

5.1.2.2 OcudrTrafficRateAboveMinorThreshold

Table 5-20 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.

5.1.2.3 OcudrTrafficRateAboveCriticalThreshold

Table 5-21 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.

5.1.2.4 OcudrTransactionErrorRateAbove0.1Percent

Table 5-22 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 5-22 (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.

5.1.2.5 OcudrTransactionErrorRateAbove1Percent

Table 5-23 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.

5.1.2.6 OcudrTransactionErrorRateAbove10Percent

Table 5-24 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 5-24 (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.

5.1.2.7 OcudrTrafficRateAboveCriticalThreshold

Table 5-25 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.

5.1.2.8 OcudrTrafficRateAboveMajorThreshold

Table 5-26 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.

5.1.2.9 OcudrTrafficRateAboveMinorThreshold

Table 5-27 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 5-27 (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.

5.1.2.10 OcudrTransactionErrorRateAbove0.1Percent

Table 5-28 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.

5.1.2.11 OcudrTransactionErrorRateAbove1Percent

Table 5-29 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.

5.1.2.12 OcudrTransactionErrorRateAbove10Percent

Table 5-30 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.

5.1.2.13 OcudrTransactionErrorRateAbove25Percent

Table 5-31 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 OcnrfTransactionErrorRateAbove50Percent 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.

5.1.2.14 OcudrTransactionErrorRateAbove50Percent

Table 5-32 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.

5.1.2.15 OcudrXFCCValidationFailureAbove10Percent

Table 5-33 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.

5.1.2.16 OcudrXFCCValidationFailureAbove20Percent

Table 5-34 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.

5.1.2.17 OcudrXFCCValidationFailureAbove50Percent

Table 5-35 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.

5.1.2.18 OcudrOverload60Percent

Table 5-36 OcudrOverload60Percent

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.

5.1.2.19 OcudrOverload75Percent

Table 5-37 OcudrOverload75Percent

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.

5.1.2.20 OcudrOverload80Percent

Table 5-38 OcudrOverload80Percent

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.

5.1.2.21 OcudrOverload90Percent

Table 5-39 OcudrOverload90Percent

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.

5.1.2.22 SLFSucessTxnDefaultGroupIdRateAbove1Percent

Table 5-40 SLFSucessTxnDefaultGroupIdRateAbove1Percent

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.

5.1.2.23 SLFSucessTxnDefaultGroupIdRateAbove10Percent

Table 5-41 SLFSucessTxnDefaultGroupIdRateAbove10Percent

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 5-41 (Cont.) SLFSucessTxnDefaultGroupIdRateAbove10Percent

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.

5.1.2.24 SLFSucessTxnDefaultGroupIdRateAbove25Percent

Table 5-42 SLFSucessTxnDefaultGroupIdRateAbove25Percent

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.

5.1.2.25 SLFSucessTxnDefaultGroupIdRateAbove50Percent

Table 5-43 SLFSucessTxnDefaultGroupIdRateAbove50Percent

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 5-43 (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.

5.1.2.26 OcudrDiameterCongestionCongestedState

Table 5-44 OcudrDiameterCongestionCongestedState

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	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.

5.1.2.27 OcudrDiameterCongestionDocState

Table 5-45 OcudrDiameterCongestionDocState

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	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.

5.1.2.28 DRProvServiceOverload60Percent

Table 5-46 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.

5.1.2.29 DRProvServiceOverload75Percent

Table 5-47 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.

5.1.2.30 DRProvServiceOverload80Percent

Table 5-48 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.

5.1.2.31 DRProvServiceOverload90Percent

Table 5-49 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.

5.1.2.32 Diameter-Gateway pod congestion Danger of congestion state

Table 5-50 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	occnp_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.

5.1.2.33 Diameter-Gateway pod CONGESTED state

Table 5-51 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	occnp_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

5.1.2.34 OcudrProvisioningTrafficRateAboveMajorThreshold

Table 5-52 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 5-52 (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 `UDR_Alertrules.yaml`. 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 recieving 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.

5.1.2.35 OcudrProvisioningTrafficRateAboveCriticalThreshold

Table 5-53 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 5-53 (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 `UDR_Alertrules.yaml`. 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 recieving 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.

5.1.2.36 OcudrProvisioningTransactionErrorRateAbove25Percent

Table 5-54 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 OcnrfTransactionErrorRateAbove50Percent 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.

5.1.2.37 OcudrProvisioningTransactionErrorRateAbove50Percent

Table 5-55 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.

5.1.2.38 PVCFullForSLFExport

Table 5-56 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.

5.1.2.39 FailedExtractForSLFExport

Table 5-57 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 5-57 (Cont.) FailedExtractForSLFExport

Field	Details
Recommended Actions	Check logs for failure. The alert will be cleared when the export job succeeds next time.

5.1.2.40 BulkImportTransferInFailed

Table 5-58 BulkImportTransferOutFailed

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.

5.1.2.41 ExportToolTransferOutFailed

Table 5-59 ExportToolTransferOutFailed

Field	Details
Description	Transfer-in failed for export-tool
Summary	Transfer-in 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.

5.1.2.42 BulkImportTransferOutFailed

Table 5-60 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.

5.1.2.43 PVCFullForXMLBulkImport

Table 5-61 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.

5.1.2.44 PVCFullForBulkImport

Table 5-62 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 5-62 (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, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

5.1.2.45 OperationalStatusCompleteShutdown

Table 5-63 OperationalStatusCompleteShutdown

Field	Details
Description	Operational state is control shutdown
Summary	Operational state is control shutdown
Severity	Critical
Condition	Alert will be raised if the opertational state of the UDR, SLF, or EIR is COMPLETE_SHUTDOWN
Metric Used	nudr_config_operational_status{kubernetes_namespace="ocudr"}==1
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, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

5.1.2.46 NFScoreCalculationFailed

Table 5-64 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	nfscore{kubernetes_namespace="ocudr" ,factor=~"successTPS|signallingConnections|serviceHealth|replicationHealth|localityPreference|bulkImport|bulkExport",calculatedStatus="failed"}
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, udr_rest_failure_response_total. 2. Contact My Oracle Support, if guidance is required.

5.1.2.47 PVCFullForEXMLExport

Table 5-65 PVCFullForEXMLExport

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.

5.1.2.48 EXMLExportFailed

Table 5-66 EXMLExportFailed

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 EXML 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.

5.1.2.49 IngressgatewayPodProtectionDocState

Table 5-67 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.

5.1.2.50 IngressgatewayPodProtectionCongestedState

Table 5-68 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.

5.1.2.51 RetryNotificationRecordsMaxLimitExceeded

Table 5-69 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.

5.1.2.52 UserAgentHeaderNotFoundMorethan10PercentRequest

Table 5-70 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.

5.1.2.53 EgressGatewayJVMBufferMemoryUsedAboveMinorThreshold

Table 5-71 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.*"}) >= 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.

5.1.2.54 EgressGatewayJVMBufferMemoryUsedAboveMajorThreshold

Table 5-72 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.

5.1.2.55 EgressGatewayJVMBufferMemoryUsedAboveCriticalThreshold

Table 5-73 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.

5.1.2.56 NudrDiameterGatewayDown

Table 5-74 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	absent(up{container="nudr-diam-gateway",namespace="ocudr"}) or up{container="nudr-diam-gateway",namespace="ocudr"} == 0

Recommended Actions	This alert is cleared when the NudrDiamGateway service is available.
	Steps:
	1. • Run the following command to check the orchestration logs of appinfo service and check for liveness or readiness probe failures.
	`kubectl get po -n <namespace>`
	• Run the following command using the full name of the pod that is not running.
	`kubectl describe pod <specific desired full pod name> -n <namespace>`
	2. Refer the application logs on Kibana and filter based on the appinfo service names. Check for `ERROR WARNING` logs related to thread exceptions.
	3. Perform the resolution steps depending on the reason for failure.
	4. Contact My Oracle Support, if guidance is required.
	Note: Use CNC NF Data Collector tool for capturing logs. For more information, see *Oracle Communications Cloud Native Core, Network Function Data Collector User Guide*.

5.1.2.57 DiameterPeerConnectionsDropped

Table 5-75 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 5-75 (Cont.) DiameterPeerConnectionsDropped

Field	Details
Metric Used	sum(ocudr_diam_conn_network{origHost=~".*CHI.*",container="nudr-diam-gateway",namespace="ocudr"} or vector(0))< 2 or sum(ocudr_diam_conn_network{origHost=~".*IND.*",container="nudr-diam-gateway",namespace="ocudr"} or vector(0)) < 2 or (sum(ocudr_diam_conn_network{origHost=~".*CHI.*",container="nudr-diam-gateway",kubernetes_namespace="ocudr"} or vector(0)) + sum(ocudr_diam_conn_network{origHost=~".*IND.*",container="nudr-diam-gateway",namespace="ocudr"}) or vector(0)) < 5

Recommended Actions	This alert is cleared when the NudrDiamGateway service is available.
	Steps:
	1. • Run the following command to check the orchestration logs of appinfo service and check for liveness or readiness probe failures.
	`kubectl get po -n <namespace>`
	• Run the following command using the full name of the pod that is not running.
	`kubectl describe pod <specific desired full pod name> -n <namespace>`
	2. Refer the application logs on Kibana and filter based on the appinfo service names. Check for `ERROR WARNING` logs related to thread exceptions.
	3. Perform the resolution steps depending on the reason for failure.
	4. Contact My Oracle Support, if guidance is required.
	Note: Use CNC NF Data Collector tool for capturing logs. For more information, see *Oracle Communications Cloud Native Core, Network Function Data Collector User Guide*.

5.1.2.58 IGWSignallingPodProtectionDOCState

Table 5-76 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	sum({namespace="ocudr",container="ingressgateway-sig"}) by (pod) == 2

Table 5-76 (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, `oc_ingressgateway_congestion_system_state`. 2. Contact My Oracle Support, if guidance is required.

5.1.2.59 IGWSignallingPodProtectionCongestedState

Table 5-77 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	sum(oc_ingressgateway_congestion_system_state{namespace="ocudr" ,container="ingressgateway-sig"}) by (pod) == 3
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, `oc_ingressgateway_congestion_system_state`. 2. Contact My Oracle Support, if guidance is required.

5.1.2.60 IGWSignallingPodProtectionByRateLimitRejectedRequest

Table 5-78 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 5-78 (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.