

Oracle® Communications

Cloud Native Core, Network Exposure Function User Guide



Release 24.2.2
F97919-04
September 2025

ORACLE®

Copyright © 2021, 2025, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

Contents

1	Introduction	
1.1	Overview	1
1.2	References	3
2	NEF Architecture	
3	NEF Features	
3.1	Support for MSISDN-Less MO-SMS	1
3.2	Converged Charging Support for NEF	5
3.3	Support for Device Trigger	10
3.4	API Invoker Onboarding and Offboarding	12
3.5	Support for Model D Communication	14
3.6	Support for Security Token Generation	20
3.7	Monitoring Event Service in NEF	22
3.7.1	Converged SCEF NEF Model for Monitoring Event	31
3.8	GMLC Based Location Monitoring	34
3.9	CAPIF Event Management	42
3.10	Support for AF Session with QoS	45
3.10.1	Converged SCEF NEF for QoS	51
3.11	Support for Georedundancy	54
3.11.1	Adding a Site to an Existing NEF Deployment	58
3.11.2	Removing a Site to from an Existing Georedundant Deployment	59
3.12	Converged SCEF-NEF	60
3.13	Support for Application Function Influence on Traffic Routing	61
3.14	Automated Test Suite Support	62
3.15	Support for Kubernetes Resource	62
3.15.1	Network Policies	62
4	Configuring Network Exposure Function using the CNC Console	
4.1	Support for Multicluster Deployment	1
4.2	CNC Console Interface	2

4.2.1	Configuring NEF Features	4
4.2.1.1	Configuring Log Level for Services	5
4.2.1.2	Configuring AF Service ID Mapping	6
4.2.1.3	Configuring QoS Reference Profile	9
4.2.1.4	Configuring AF ID Mapping	10
4.2.1.5	Configuring GMLC Options	11
4.2.1.6	Configuring QoS Options	12
4.2.1.7	Configuring Short or Long Code for MSISDNless MO SMS	13
4.2.1.8	Configuring SCS Short Messaging Entity	15
4.2.1.9	Configuring PLMN ID Mapping	15
4.2.1.10	Configuring TAI Mapping	16
4.2.1.11	Configuring ECGI Mapping	17
4.2.1.12	Configuring NCGI Mapping	19
4.2.1.13	Configuring GNBID Mapping	20
4.2.1.14	Configuring GlobalRANNodeID Mapping	21
4.2.1.15	Configuring GeoZoneIdToSpatialValidity Mapping	22
4.2.1.16	Viewing Global System Config	24
4.2.1.17	Configuring Gateway	25
4.2.1.18	Viewing cnDBTier APIs in CNC Console	27
4.2.2	Configuring CAPIF Features	32
4.2.2.1	Configuring Log Level for Services	32
4.2.2.2	Configuring Invoker Access Token	33
4.2.2.3	Configuring Discovery Group	34
4.2.2.4	Configuring Invoker Pre-provisioning	35

5 Configuring NEF

6 NEF Metrics

6.1	NEF OCI Metrics Dashboard	1
6.2	Dimension Description	1
6.3	NEF Metrics	2
6.3.1	API Router Metrics	2
6.3.2	CCF Client Metrics	4
6.3.3	ME Service Metrics	5
6.3.4	QoS Service Metrics	9
6.3.5	5GC Agent Service Metrics	13
6.3.6	Expiry Auditor Service Metrics	15
6.3.7	Traffic Influence Metrics	16
6.3.8	Converged SCEF-NEF Metrics	18
6.3.9	Device Trigger Metrics	19

6.3.10	Diameter Gateway Metrics	23
6.3.11	MSISDNless MO SMS Metrics	24
6.4	CAPIF Metrics	25
6.4.1	API Manager Metrics	26
6.4.2	AF Manager Metrics	27
6.4.3	Event Manager Metrics	28
6.5	Ingress Gateway Metrics	29
6.6	Egress Gateway Metrics	35

7 Alerts

7.1	List of Alerts	1
7.1.1	NEF Alerts	1
7.1.1.1	System Level Alerts	2
7.1.1.2	Application Level Alerts	26
7.1.2	CAPIF Alerts	41
7.1.2.1	System Level Alerts	42
7.1.2.2	Application Level Alerts	61
7.2	Configuring Alerts	61
7.2.1	Configuring Alert Manager for SNMP Notifier	64
7.3	NEF Alert Configuration in OCI	66
7.3.1	Configuring NEF Alerts for OCI	67

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 lists the acronyms and the terminologies used in the document:

Table Acronyms and Terminologies

Acronym	Description
3GPP	3rd Generation Partnership Project
5GC	5G Core Network
5GS	5G System
AF	Application Function
API	Application Programming Interface
CAPIF	Oracle Communications Common API Framework
CNC	Oracle Communications Cloud Native Core
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
ECUR	Event Charging with Unit Reservation
FQDN	Fully Qualified Domain Name
GPSI	Generic Public Subscription Identifier
GMLC	Gateway Mobile Location Center
HA	High Availability
IEC	Immediate Event Charging
IGW	Internet Gateway
IMSI	International Mobile Subscriber Identity
K8s	Kubernetes
ME	Monitoring Events
MSISDN	Mobile Station International Subscriber Directory Number
NEF	Oracle Communications Cloud Native Core, Network Exposure Function
NF	Network Function
NRF	Oracle Communications Cloud Native Core, Network Repository Function
NSSF	Oracle Communications Cloud Native Core, Network Slice Selection Function
OAM	Operations, Administration, and Maintenance
OCI	Oracle Cloud Infrastructure
OKE	Oracle Kubernetes Engine
PDB	Pod Disruption Budget
QoS	Quality of Service
SBA	Service Based Architecture
SBI	Service Based Interface
SCEF	Service Capability Exposure Function
S-NSSAI	Single Network Slice Selection Assistance Information
SUPI	Subscription Permanent Identifier
TPS	Traffic Per Second
UDM	Unified Data Management

Table (Cont.) Acronyms and Terminologies

Acronym	Description
URI	Uniform Resource Identifier

What's New in This Guide

This section introduces the documentation updates for Release 24.2.x.

Release 24.2.2 - F97919-04, September 2025

Added the following metrics in the [Converged SCEF-NEF Metrics](#) and [Diameter Gateway Metrics](#) sections:

- ocnef_diamgw_diam_req_total
- ocnef_diamgw_diam_resp_total
- ocnef_diamgw_http_req_total
- ocnef_diamgw_http_resp_total

Removed the following metrics from the [Converged SCEF-NEF Metrics](#) section:

- ocnef_diamgw_request
- ocnef_diamgw_response
- ocnef_diamgw_backend_request
- ocnef_diamgw_backend_response

Updated the release number to 24.2.2 in the entire document.

Release 24.2.1 - F97919-03, October 2024

Updated the release number to 24.2.1 in the entire document.

Release 24.2.0 - F97919-02, August 2024

Added OID information for NEF and CAPIF alerts in [NEF Alerts](#) and [CAPIF Alerts](#) sections.

Release 24.2.0 - F97919-01, July 2024

The following changes are made in this document:

- Added the following sections as part of Integration with CNC Console feature:
 - [Configuring PLMN ID Mapping](#)
 - [Configuring TAI Mapping](#)
 - [Configuring ECGI Mapping](#)
 - [Configuring NCGI Mapping](#)
 - [Configuring GNBID Mapping](#)
 - [Configuring GlobalRANNodeID Mapping](#)
 - [Configuring GeoZoneIdToSpatialValidity Mapping](#)
 - [Configuring Log Level for Services](#)
 - [Configuring Invoker Access Token](#)
 - [Configuring Discovery Group](#)
 - [Configuring Invoker Pre-provisioning](#)
- Updated the following sections as part of Integration with CNC Console feature:

- [Support for Multicloud Deployment](#)
- [Configuring CAPIF Features](#)
- Updated Type from Quantile to Histogram for the following metrics in [NEF Metrics](#):
 - ocnef.aef.apirouter.latency
 - ocnef.aef.apirouter.backend.latency
 - ocnef_msisdnless_mo_sms_srv_latency
 - ocnef_chf_qos_latency
- Removed Time dimension from [Table 6-99](#) table.
- Removed **NEF Configuration Using REST APIs** chapter.
- The following changes are made as part of the Deployment in OCI using OCI Adaptor feature:
 - Added the OCI acronym in the [Acronyms](#) section.
 - Added the *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide* document in the [References](#) section.
 - Updated the [Introduction](#) section with supported deployment platforms.
 - Added the following sections to configure metrics and alerts for OCI:
 - * [NEF Alert Configuration in OCI](#)
 - * [Configuring NEF Alerts for OCI](#)
 - * [NEF OCI Metrics Dashboard](#)
 - Added the following alerts as part of Deployment in OCI feature:
 - * [OccapifTotalExternalIngressTrafficRateAboveMinorThreshold](#)
 - * [OccapifTotalNetworkIngressTrafficRateAboveMinorThreshold](#)
 - * [OccapifTotalExternalIngressTrafficRateAboveMajorThreshold](#)
 - * [OccapifTotalNetworkIngressTrafficRateAboveMajorThreshold](#)
 - * [OccapifTotalExternalIngressTrafficRateAboveCriticalThreshold](#)
 - * [OccapifTotalNetworkIngressTrafficRateAboveCriticalThreshold](#)
 - * [OccapifExternalIngressTransactionErrorRateAboveZeroPointOnePercent](#)
 - * [OccapifNetworkIngressTransactionErrorRateAboveZeroPointOnePercent](#)
 - * [OccapifExternalIngressTransactionErrorRateAbove1Percent](#)
 - * [OccapifNetworkIngressTransactionErrorRateAbove1Percent](#)
 - * [OccapifExternalIngressTransactionErrorRateAbove10Percent](#)
 - * [OccapifNetworkIngressTransactionErrorRateAbove10Percent](#)
 - * [OccapifExternalIngressTransactionErrorRateAbove25Percent](#)
 - * [OccapifNetworkIngressTransactionErrorRateAbove25Percent](#)
 - * [OccapifExternalIngressTransactionErrorRateAbove50Percent](#)
 - * [OccapifNetworkIngressTransactionErrorRateAbove50Percent](#)
 - * [OcnefTotalExternalIngressTrafficRateAboveMinorThreshold](#)
 - * [OcnefTotalFivegcIngressTrafficRateAboveMinorThreshold](#)

-
- * [OcnefTotalExternalIngressTrafficRateAboveMajorThreshold](#)
 - * [OcnefTotalFivegcIngressTrafficRateAboveMajorThreshold](#)
 - * [OcnefTotalExternalIngressTrafficRateAboveCriticalThreshold](#)
 - * [OcnefTotalFivegcIngressTrafficRateAboveCriticalThreshold](#)
 - * [OcnefExternalIngressTransactionErrorRateAboveZeroPointOnePercent](#)
 - * [OcnefFivegcIngressTransactionErrorRateAboveZeroPointOnePercent](#)
 - * [OcnefExternalIngressTransactionErrorRateAbove1Percent](#)
 - * [OcnefFivegcIngressTransactionErrorRateAbove1Percent](#)
 - * [OcnefExternalIngressTransactionErrorRateAbove10Percent](#)
 - * [OcnefFivegcIngressTransactionErrorRateAbove10Percent](#)
 - * [OcnefExternalIngressTransactionErrorRateAbove25Percent](#)
 - * [OcnefFivegcIngressTransactionErrorRateAbove25Percent](#)
 - * [OcnefExternalIngressTransactionErrorRateAbove50Percent](#)
 - * [OcnefFivegcIngressTransactionErrorRateAbove50Percent](#)
 - Removed the following alerts as part of Deployment in OCI feature:
 - * OccapifTotalIngressTrafficRateAboveMinorThreshold
 - * OccapifTotalIngressTrafficRateAboveMajorThreshold
 - * OccapifTotalIngressTrafficRateAboveCriticalThreshold
 - * OccapifTransactionErrorRateAbove0.1Percent
 - * OccapifTransactionErrorRateAbove1Percent
 - * OccapifTransactionErrorRateAbove10Percent
 - * OccapifTransactionErrorRateAbove25Percent
 - * OccapifTransactionErrorRateAbove50Percent
 - * OcnefTotalIngressTrafficRateAboveMinorThreshold
 - * OcnefTotalIngressTrafficRateAboveMajorThreshold
 - * OcnefTotalIngressTrafficRateAboveCriticalThreshold
 - * OcnefTransactionErrorRateAbove0.1Percent
 - * OcnefTransactionErrorRateAbove1Percent
 - * OcnefTransactionErrorRateAbove10Percent
 - * OcnefTransactionErrorRateAbove25Percent
 - * OcnefTransactionErrorRateAbove50Percent
 - Added the [Configuring Alert Manager for SNMP Notifier](#) section as part of Support for CNC Top Level MIB in NEF feature.
 - Added a note on guidelines on mandatory and optional parameters validation in NEF in [NEF Features](#) section.
 - Added a note on Edit GMLC operation in [Configuring GMLC Options](#) section.

1

Introduction

This document provides information about the role of Oracle Communications Cloud Native Core, Network Exposure Function (NEF) in 5G Service Based Architecture (SBA) and how to configure and use the NEF functionality and services.

NEF installation is supported over the following platforms:

- Oracle Communications Cloud Native Core, Cloud Native Environment (CNE) Oracle Cloud Infrastructure (OCI)
- For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

1.1 Overview

NEF is a key component of the 5G Service Based Architecture. It provides a platform to securely expose the network services and capabilities offered by the 5G Network Functions (NFs) to either third-party applications or the internal Application Functions (AFs). Located between the 5G core network and third-party applications or AFs, NEF enables the external application administrators to customize the network for providing innovative services to their end-users. The applications communicate through NEF to access the internal data of the 5G core network.

Note

The performance and capacity of the NEF system may vary based on the call model, feature or interface configuration, and underlying CNE/OCI and hardware environment.

NEF performs the following functions:

- Facilitates robust and secure exposure of network services, such as voice, data connectivity, charging, subscriber data, IoT, and so on to trusted third-party applications or AFs
- Provides programmable environment access of 5G network to both internal and external application administrators through a set of northbound RESTful APIs
- Enables AF to securely provide information to 3GPP network to authenticate, authorize, and assist in throttling the AF
- Translates the information received from the AF to the internal 3GPP NFs, and vice versa
- Provides support to expose information collected from other 3GPP NFs to the AF
- Monitors User Equipment (UEs) related events present in the 5G system and makes the event information available for external exposure. For example, monitoring of user location and services.

NEF interacts with different applications and the 5G core network. It performs the above functions through the following services:

- NEF 5GC Agent
- NEF CAPIF Core Function (CCF) Client
- NEF Expiry Auditor
- NEF Monitoring Events
- NEF Quality of Service
- NEF Traffic Influence
- NEF API Router
- NEF APD Manager
- NEF Diameter Gateway
- CAPIF AF Manager
- CAPIF API Manager
- CAPIF Event Manager
- Device Trigger
- MSISDNless MO SMS

For a detailed description of each service, see [NEF Architecture](#).

NEF Availability

Oracle Communications Cloud Native Core (CNC) NEF availability is dependent on many factors. NEF applications are designed to achieve 99.999% availability, according to the applicable Telecommunications Industry Association TL9000 standards, with the following deployment requirements:

- Deploy on a Cloud Native Environment with at least 99.999% Availability.
- Deploy with $n + k$ application redundancy, where k is greater than or equal to one.
- Maintain production software within $n-3$ software releases, where n is the current general availability release.
- Apply bug fixes, critical patches, and configuration recommendations provided by Oracle promptly.
- Maintain fault recovery procedures external to the applications for the reconstruction of lost or altered files, data, programs, or Cloud Native environment.
- Install, configure, operate, and maintain NEF as per Oracle's applicable installation, operation, administration, and maintenance specifications.
- Maintain an active support contract and provide access to the deployed NEF and your personnel to assist Oracle in addressing any outage.

NEF availability is measured for each calendar year and is calculated as follows:

Table 1-1 Measuring NEF Availability

Availability	Description
Planned Product Availability	(Product available time in each month) less (Excluded Time (defined below) in each month).
Actual Product Availability	(Planned Product Availability) less (any Unscheduled Outage)

Table 1-1 (Cont.) Measuring NEF Availability

Availability	Description
Product Availability Level	(Actual Product Availability across all Production instances divided by Planned Product Availability across all Production instances) x 100

Note**Excluded Time** means:

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

1.2 References

Following are the reference documents:

- *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*
- *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*
- [3GPP Technical Specification 29.222, Common API Framework for 3GPP, Release 16](#)
- [3GPP Technical Specification 29.571, Common Data Types for Service Based Interfaces](#)
- [3GPP Technical Specification 23.222, Common API Framework for 3GPP Northbound APIs](#)

2

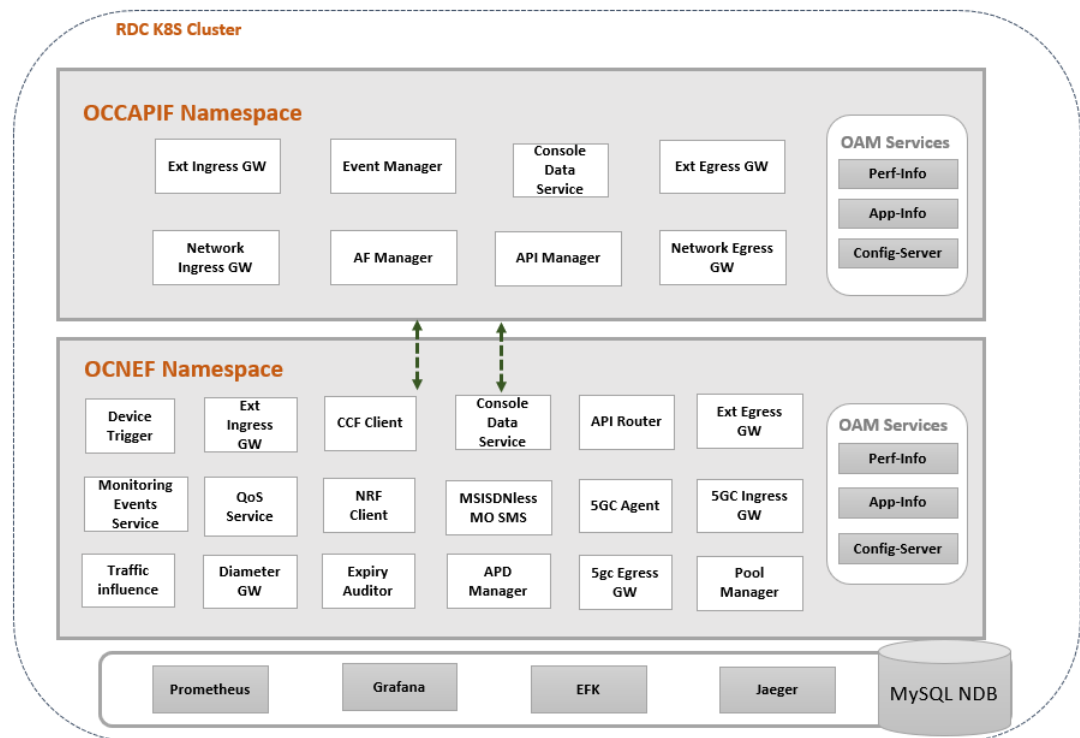
NEF Architecture

Oracle Communications Cloud Native Core, Network Exposure Function (NEF) is a cloud native application. The application consists of the following components running on separate namespaces in the cloud native environment:

- **CAPIF:** The service interface between NEF and the external third-party applications or internal Application Functions (AFs).
CAPIF is a 3GPP defined secured framework to expose network service interfaces. It enables the API invokers (external applications) to discover and communicate with service APIs of the API provider (NEF). This framework manages API security, logging of events, auditing capability, multiple service exposure, policy based routing, dynamic routing of information, and so on.
- **NEF:** The core component that runs the business logic of NEF. It consists of various services that interacts with the CAPIF and performs the core functionality of NEF.

The NEF architecture is grouped into different components. Each component consists of multiple microservices that perform the core functionality. The following diagram represents the NEF architecture:

Figure 2-1 NEF Architecture



Components of the NEF Architecture

NEF consists of the following components and services:

- Kubernetes cluster hosting Docker containers and Calico networking
- Optional CNE services to support operations of NEF
- CAPIF services:
 - **API Manager:** Responsible for managing the registration and publish functionality of NEF. The API Provider Domain (APD) manager service handles all the transactions related to NEF using the package services of the CAPIF that are useful for the core functionality of NEF.
 - **AF Manager:** Responsible for the secured interactions between API Invokers (external applications) and API Provider (NEF). This service facilitates the following tasks:
 1. API Invoker onboarding and offboarding by establishing a communication between API Invoker and CAPIF.
 2. Security Context creation. The API invoker negotiates and obtains the information about service API security method from the AF manager service.
 - **Console Data Service:** Responsible for all the configuration APIs and all CAPIF configurations managed through CNC Console GUI.
 - **Event Manager:** Manages the subscription, unsubscription, and notification for all the events supported by CAPIF. This service facilitates AFs and other NEF services to subscribe to CAPIF specific event notifications, receive notifications about the subscribed events, and unsubscribe from the notifications.
 - **External Ingress Gateway:** Acts as a gateway for all the HTTP requests towards CAPIF from external applications. This service provides security and load balancing functionality to manage and control the incoming traffic.
 - **External Egress Gateway:** Acts as a gateway for all the HTTP requests from CAPIF to external applications. This service provides security and load balancing functionality to manage and control the outgoing traffic.
 - **Network Ingress Gateway:** Acts as a gateway for all the HTTP requests towards CAPIF from NEF.
 - **Network Egress Gateway:** Acts as a gateway for all the HTTP requests from CAPIF to NEF.
- NEF services:
 - **API Router:** Responsible for validation of the OAuth token received with a service request sent from API Invoker to NEF. The API Router microservice ensures that only authenticated requests are directed to NEF.
 - **Monitoring Event Service:** Responsible for monitoring of specific events in 3GPP system, as requested by the AFs. It communicates with the other applications to get the specific information and reports it to the respective AF.
The ME service performs event monitoring function that includes monitoring event configuration, monitoring event report, and network initiated notification of monitoring event cancellation.
 - **Quality of Service(QoS) Service:** Responsible for provisioning of QoS specifications in 3GPP system, as requested by the AFs. It communicates with the other NFs to set up data sessions with the required QoS.
The NEF QoS service facilitates the provisioning capability to AFs for sending the quality related parameters to NEF and receive the corresponding notifications.

The service allows AFs to set up a session with required QoS and priority handling that includes AF session subscription, AF session deletion, and QoS event notification.

- **Expiry Auditor:** Detects and processes the expired subscription records for the ME service.
- **APD Manager:** Responsible for monitoring the site status in a georedundant deployment using the cnDBTier REST APIs. This service monitors the notifications received from Network Repository Function (NRF) and tracks the health of all the NEF instances in a georedundant deployment.
- **Traffic Influence:** Handles all the traffic influence subscription requests, validates and manages the data associated with these requests.
- **Diameter Gateway:** Acts as a gateway for all Diameter traffic to NEF Solution. The diameter interface supported here is T6x.
- **Device Trigger:** The Device Trigger feature enables an Application Function (AF) to notify a particular User Equipment (UE) by sending a device trigger request through 5G core (5GC) to perform application-specific tasks such as initiating communication with AF. This is required when the AF does not hold information of IP address for the UE or if the UE is not reachable.
- **External Ingress Gateway:** Acts as a gateway for all ingress traffic originating from the AFs towards NEF.
- **External Egress Gateway:** Acts as a gateway for all egress traffic originating from NEF solution to AFs.
- **5GC Ingress Gateway:** Acts as a gateway for all ingress traffic originating from the 5G network functions towards NEF.
- **5GC Egress Gateway:** Acts as a gateway for all egress traffic originating from NEF solution to the 5G network functions.
- **5GC Agent:** Responsible for generating and sending the service API requests to 5G NFs. It also receives event notifications from the NFs based on the subscription created by NEF.
- **NRF Client Service:** Integrates with NRF for NEF registration, discovery, and service status or load related information, along with application and performance information services. NRF discovery helps in the on-demand discovery of network functions. NRF management helps in the autonomous discovery of network functions.
- **Pool Manager:** Responsible for allocating SCEF referenceId ranges for each of the features handling microservices (Monitoring Events, Device Trigger) across sites and managing membership state for each of these allocations.
- **Console Data Service:** Responsible for all the configuration APIs and all NEF configurations managed through CNC Console GUI.
- **MSISDNless MO SMS:** Responsible for allowing NEF to deliver the MSISDN-less MO-SMS notification message from Short Message Service - Service Center (SMSSC) to Application Function (AF).
- **Database:** This is the MySQL NDB storage engine.

3

NEF Features

This chapter explains the Oracle Communications Cloud Native Core, Network Exposure Function features.

Note

- The performance and capacity of the NEF system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.
- Following are the guidelines on mandatory and optional parameters validation in NEF as per the specification 3GPP TS 29.500, section 5.2.7.4:
 - Every request message is checked for presence of all the mandatory parameters. If a mandatory parameter is missing, that request is rejected.
 - NEF validates only those parameters (mandatory or optional) coming from external entity like AF that impacts NEF processing or call flow (involves making some decision, db lookup, creating subscription, and so on).
 - For all remaining parameters, NEF acts as pass-through and will not validate. If these parameters impact any processing by subsequent NFs in the call flow, NEF expects error to come from those NFs and that will be propagated to AF.
 - NEF will not validate any messages that are coming from trusted 5GC entities (like internal NFs), it will only rely on 5GC entities to send correct messages or parameters. In any case, if a 5GC entity sends some wrong parameter then that results in failed lookups at NEF (example, invalid correlation id) and a appropriate error response gets generated.

3.1 Support for MSISDN-Less MO-SMS

The Support for MSISDN-Less-MO-SMS feature enables NEF to deliver the MSISDN-less MO-SMS notification message from Short Message Service - Service Center (SMSSC) to Application Function (AF).

With this feature, user equipment (UE) can send messages to AF without using Mobile Station International Subscriber Directory Number (MSISDN) through T4 interface, which is an interface between SMS-SC and NEF. NEF uses the Nnef_MSISDN-Less-MO-SMS API to send UE messages to AF.

As MSISDN contains critical subscriber information, UE messages are sent to AF without MSISDN to:

- protect subscribers' data.
- expand the capabilities of IoT devices.
- enhance the security and efficiency of messaging across various applications.

UEs are pre-configured with the Service Center address that is mapped with SMS-SC. If UE contains multiple Generic Public Subscription Identifier (GPSIs) associated with the same

IMSI, the GPSI that is associated with an SMS can be determined from the UE's IMSI and the Application Port ID value in the TP-User-Data field. NEF obtains the GPSI by querying UDM using the IMSI and application port ID.

Note

- AF addresses are pre-configured in NEF.
- The Supported-Features attribute in N33 interface is set to 0, which implies that there are no features defined towards AF.

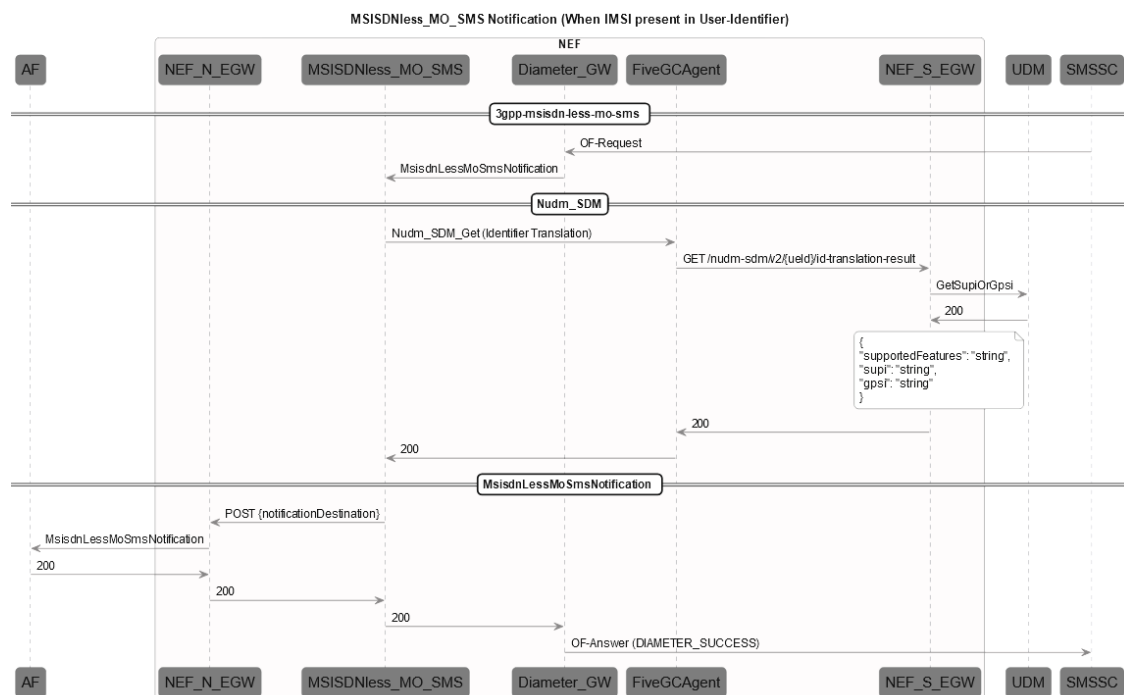
Call Flows

This section describes the call flow of the MSISDN-Less-MO-SMS feature.

Notification Flow

The following call flow represents how SMS-SC communicates with AF through Diameter Interface.

Figure 3-1 MSISDN-Less MO-SMS Notification Call Flow



1. SMSSC sends the OF-Request message to NEF.
Sample Notification Message Request

```

SUCCESS: Diameter Message: OFR
Version: 1
Msg Length: 296
Cmd Flags: REQ,PXY
Cmd Code: 8388645
App-Id: 16777313
  
```



```

Hop-By-Hop-Id: 2196095456
End-To-End-Id: 1868559833
Origin-Host (264,M,l=17) = T6-client
Origin-Realm (296,M,l=22) = mme.oracle.com
Session-Id (263,M,l=17) = session123
User-Identifier (3102,VM,v=10415,l=40) =
    External-Identifier (3111,VM,v=10415,l=27) = 1234@oracle.com
SC-Address (3300,VM,v=10415,l=22) = 8967452301
SM-RP-UI (3301,VM,v=10415,l=82) =
710D0B911326880736F40000A91506050412121213F7FBDD454E87CDE1B0DB357EB701
Auth-Session-State (277,M,l=12) = NO_STATE_MAINTAINED (1)
Vendor-Specific-Application-Id (260,M,l=32) =
    Vendor-Id (266,M,l=12) = 10415
    Auth-Application-Id (258,M,l=12) = 16777313
Destination-Realm (283,M,l=18) = oracle.com

```

Table 3-1 Parameters for Request Message

Attribute Name	Data Type	Description
Origin-Host	DiameterIdentity	The Origin-Host AVP identifies the endpoint that originated the Diameter message. Relay agents must not modify this AVP. The value of the Origin-Host AVP is guaranteed to be unique within a single host.
Origin-Realm	DiameterIdentity	This AVP contains the Realm of the originator of any Diameter message and must be present in all messages. This AVP should be placed as close to the Diameter header as possible.
Session-Id	UTF8String	The Session-Id AVP is used to identify a specific session. All messages pertaining to a specific session must include only one Session-Id AVP and the same value must be used throughout the life of a session. Each Session-Id is eternally unique. When present, the Session-Id should appear immediately following the Diameter Header.
User-Identifier	Grouped	The User-Identifier AVP contains the different identifiers used by the UE.
User-Identifier.External-Identifier	UTF8String	This information element contains External-Identifier.
User-Identifier.User-Name	UTF8String	This information element contains IMSI.
User-Identifier.MSISDN	OctetString	This information element contains MSISDN.
SC-Address	OctetString	The SC-Address AVP contains the E164 number of the SMS-SC and is encoded as TBCD-string. This AVP shall not include leading indicators for the nature of address and the numbering plan; it shall contain only the TBCD-encoded digits of the address.
SM-RP-UI	OctetString	The SM-RP-UI contains a short message transfer protocol data unit and represents the user data field carried by the short message service relay sub-layer protocol. Its maximum length is of 200 octets. Note: For further details on SMS-SUBMIT message, refer to TS 23.040. The shortcode and applicationPort are extracted from this field. Note: The 1 0 1 destination address type (coded according to 3GPP TS 23.038 [9] GSM 7-bit default alphabet) is not supported. The data type for this is alphanumeric.
Auth-Session-State	Enumerated	The Auth-Session-State AVP specifies whether state is maintained for a particular session. The client may include this AVP in requests as a hint to the server, but the value in the server's answer message is binding.

- NEF uses the Diameter-GW microservice to translate the message and sends it to MSISDNless_MO_SMS microservice.

The MSISDNless_MO_SMS microservice matches short code in the notification message with the pre-configured short codes in the NEF microservices to obtain the AF URL. If the short code does not match, NEF returns the error response with SM-Delivery-Failure-Cause is set to UNKNOWN_SERVICE_CENTRE and Experimental-Result is set to SM_DELIVERY_FAILUR_CAUSE_EXPERIMENTAL_RESULT_CODE = 5555.

- external ID is present in the notification message, NEF sends the message to AF.
- If the external ID is absent, NEF sends the Nudm_SDM_Get (Identifier Translation) message to UDM to obtain the external ID. If UDM sends an error response, Experimental-Result is set to SM_DELIVERY_FAILUR_CAUSE_EXPERIMENTAL_RESULT_CODE = 5555 and SM-Delivery-Failure-Cause is set to USER_NOT_SC-USER. If UDM responds with success code, NEF sends the message to AF.

3. AF responds with success code, NEF responds with OF-Answer message to SMSSC.
Sample Notification Message Response

```
SUCCESS: Diameter Message: OFA
Version: 1
Msg Length: 236
Cmd Flags: PXY
Cmd Code: 8388645
App-Id: 16777313
Hop-By-Hop-Id: 2196095456
End-To-End-Id: 1868559833
  Session-Id (263,M,l=17) = sesion123
  Result-Code (268,M,l=12) = DIAMETER_SUCCESS (2001)
  Origin-Host (264,M,l=26) = ocnef-diam-gateway
  Origin-Realm (296,M,l=18) = oracle.com
  Auth-Session-State (277,M,l=12) = NO_STATE_MAINTAINED (1)
  SM-RP-UI (3301,VM,v=10415,l=82) =
710D0B911326880736F40000A91506050412121213F7FBDD454E87CDE1B0DB357EB701
  External-Identifier (3111,VM,v=10415,l=27) = 1234@oracle.com
  Auth-Session-State (277,M,l=12) = NO_STATE_MAINTAINED (1)
```

Table 3-2 Parameters for Response Message

Attribute Name	Data Type	Description
Origin-Host	DiameterIdentity	The Origin-Host AVP identifies the endpoint that originated the Diameter message. Relay agents must not modify this AVP. The value of the Origin-Host AVP is guaranteed to be unique within a single host.
Origin-Realm	DiameterIdentity	This AVP contains the Realm of the originator of any Diameter message and must be present in all messages. This AVP should be placed as close to the Diameter header as possible.
Session-Id	UTF8String	The Session-Id AVP is used to identify a specific session. All messages pertaining to a specific session must include only one Session-Id AVP and the same value must be used throughout the life of a session. Each session id is eternally unique. When present, the Session-Id should appear immediately following the Diameter Header.
Result-Code	Unsigned32	The Result-Code AVP indicates whether a particular request was completed successfully or whether an error occurred.

Table 3-2 (Cont.) Parameters for Response Message

Attribute Name	Data Type	Description
Experimental-Result	Grouped	The Experimental-Result AVP indicates whether a particular vendor-specific request was completed successfully or whether an error occurred Supported Experimental-Result: SM_DELIVERY_FAILUR_CAUSE_EXPERIMENTAL_RESULT_CODE = 5555
SM-Delivery-Failure-Cause	Grouped	The SM-Delivery-Failure-Cause AVP is of type Grouped, and contains information about the cause of the failure of a SM delivery with an optional Diagnostic information Supported SM-Delivery-Failure-Cause: when short code doesnt match -> UNKNOWN_SERVICE_CENTRE when getGpsi from udm returns error -> USER_NOT_SC-USER
Auth-Session-State	Enumerated	The Auth-Session-State AVP specifies whether state is maintained for a particular session. The client may include this AVP in requests as a hint to the server, but the value in the server's answer message is binding.
SM-RP-UI	OctetString	The SM-RP-UI contains a short message transfer protocol data uni and represents the user data field carried by the short message service relay sub-layer protocol. Its maximum length is of 200 octets.
External-Identifier	UTF8String	The External-Identifier AVP shall contain an external identifier of the UE.

Managing Support for MSISDN-Less-MO-SMS for NEF**Enable**

You can enable the MSISDN-Less-MO-SMS service by setting `msisdnlless_mo_sms` parameter to true. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure this feature using Helm parameters. You can create the Model A, B, and D communication profiles using the `custom-values.yaml` file.

For information about configuring the `msisdnllessmosms` microservice parameters, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

For information about configuring this feature through CNC Console, see *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

Observe

NEF provides the MSISDN-Less-MO-SMS service related metrics for observing AF Session with this feature. For more information about the metrics, see [Diameter Gateway Metrics](#) and [MSISDNless MO SMS Metrics](#).

3.2 Converged Charging Support for NEF

This feature facilitates NEF with charging capabilities for its northbound API invocation. This allows NEF to engage in converged charging through interaction with CHF NF using Network Charging Function (Nchf).

The Converged Charging feature allows NEF to communicate with CHF NF to perform any charging related activities. This functionality allows to gather the necessary data for various management activities, such as credit control, accounting, billing, and statistics.

Note

For further information, refer to TS 32.290 and TS 32.291 documents.

For northbound API access, NEF performs convergent charging in collaboration with CHF.

The exchange of charging data request and response between NEF and CHF is performed based on PEC, by either following Immediate Event Charging (IEC) or Event Charging with Unit Reservation (ECUR) scenarios specified in TS 32.290. NEF initiates the charging data requests towards CHF when specific conditions (chargeable events) are met.

Note

- NEF uses `Nchf_ConvergedCharging` API only for PEC scenarios.
- The selection of the CHF can be configured in NEF by relying on NRF.

Call Flows

For invocation, after creating a AS Session with Quality of Service (QoS) call flow, NEF interacts with CHF to create a Call Detail Record (CDR) of API Invocation. It then processes the QoS payload to create a CDR and then passes the CDR message to 5G Core (5GC) with necessary headers. 5GC forwards the same request to CHF and allows NEF to create a CDR message on CHF for all the successful and failed call flows.

For notification, after notifying a AS Session with QoS call flow, NEF interacts with CHF to create a CDR of API Notification. It then processes the QoS payload to create a CDR in 5GC. The actual notification message in 5GC is asynchronously passed to CHF. 5GC then creates the CDR message and forwards the same request to CHF. NEF then creates CDR for all the successfully processed as well as failed notifications. The `apiContent` parameter holds the JSON sent to AF for the notification.

Note

- This feature can be enabled through `qualityofservice.converged.pe.charging` parameter. For further information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation and Upgrade Guide*.
- The call flows are valid only if the feature is enabled.
- Currently, this feature is implemented only for QoS service API (`3gpp-as-session-with-qos`). For QoS service, the `PRODUCER_NF` is PCF for 5G and PCRF for 4G fallback. For further information on QoS call flow, refer to *Support for AF Session with QoS* section in *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

API Invocation Call Flow in Converged Charging

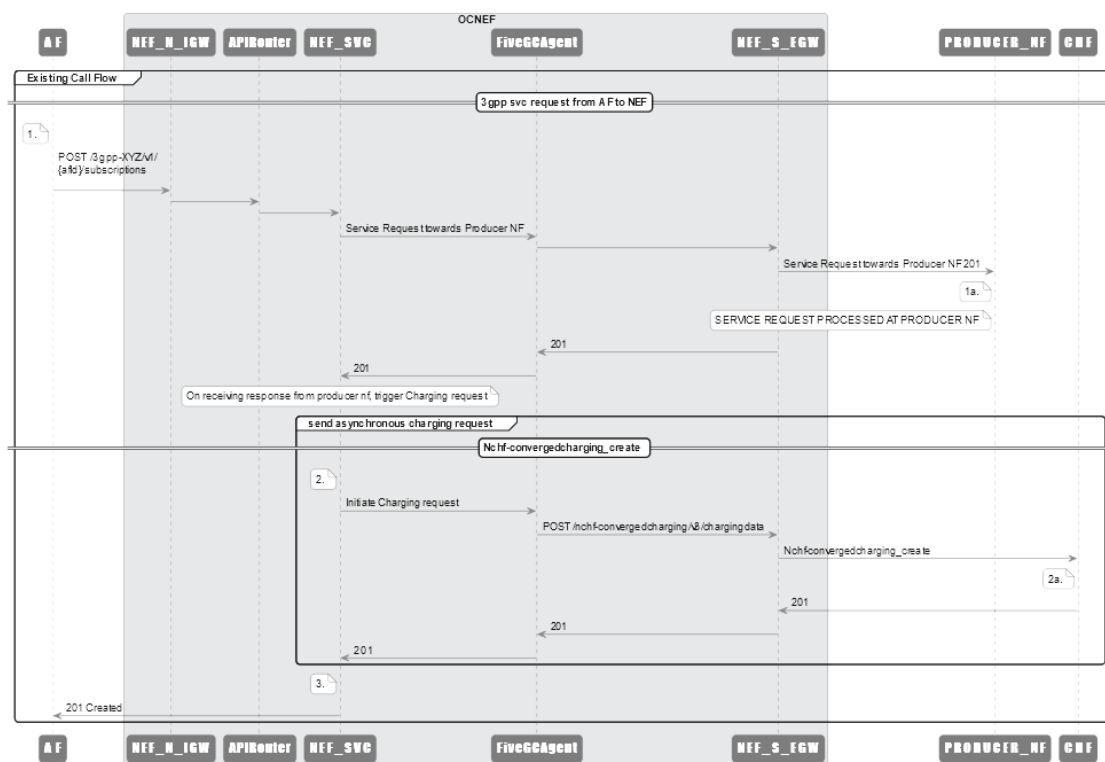
Any request triggered by AF is an invocation request. This call flow scenario represents how an NEF interacts with CHF to create a CDR of API Invocation, post creating AS Session when QoS call flow is completed.

Note

- In case of an invocation, the message reaching QoS service is the confirmation for CDR generation.
- This call flow is not applicable for GET method.

In this following scenario, once the actual request from AF is complete, NEF initiates a one time charging request towards CHF using `Nchf-convergedcharging_create` API. This creates a converged charging request at CHF, which then can be used for further billing and other charging related functionalities.

Figure 3-2 API Invocation Call Flow



The above call flow represents an AF invocation request being processed by Converged Charging scenario in NEF:

1. AF initiates a POST 3gpp-as-session-with-qos service request to northbound NEF Ingress Gateway.

Note

Currently, only the 3gpp-as-session-with-qos service API is supported.

2. After completing create request, NEF processes the payload to create a CDR (asynchronously).

3. NEF sends the request to APIRouter.
4. APIRouter validates the request and sends it to NEF service.
5. NEF service then passes the CDR message to FiveGCAGENT with the necessary headers.

Note

NEF creates CDR message on CHF for all the successful and failed call flows.

6. The request is then sent to southbound NEF Egress Gateway.
7. The message then reaches the PRODUCER_NF, where the request is processed.
8. Post processing, the PRODUCER_NF returns the response to NEF service through NEF Egress Gateway and FiveGCAGENT.
9. Once the response is received by NEF service, it initiates a charging request and a asynchronous converged post event charging request to CHF.
10. The asynchronous call reaches FiveGCAGENT, which is then translated to a charging request and sent to southbound NEF egress gateway.
11. Southbound NEF Egress Gateway sends the request to CHF.
12. CHF processes the request and sends the response to NEF Service.

API Notification Call Flow in Converged Charging

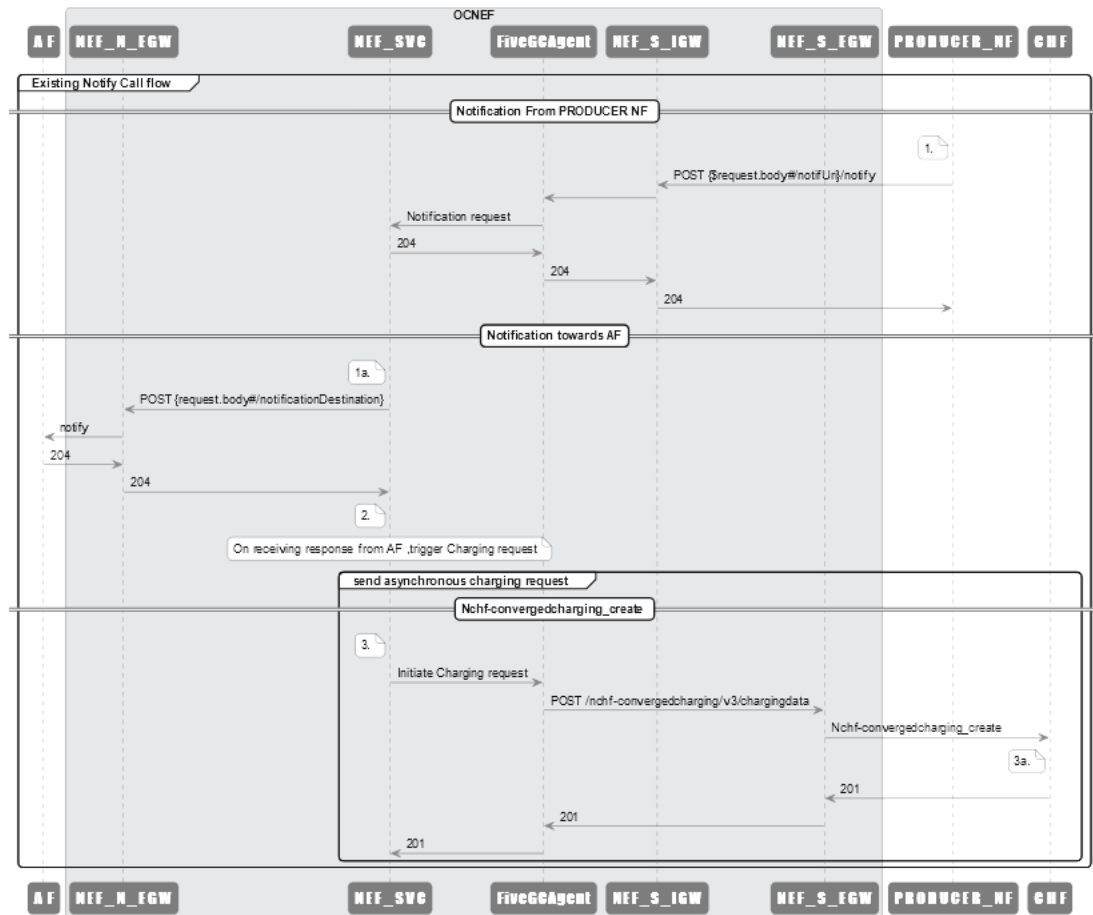
Any request sent from PRODUCER_NF is a Notification request. This call flow scenario represents how an NEF interacts with CHF to create a CDR of API Notification, post completion of notification AS Session with the QoS call flow.

Note

In case of notification, the message reaching AF is the confirmation for CDR generation.

In this following scenario, once the PRODUCER_NF sends a request towards NEF, NEF notifies the request to AF. NEF initiates a one time converged charging request towards CHF using `Nchf-convergedcharging_create` API, after the notification request is received. This creates a converged charging request at CHF, which then can be used for further billing and other charging related functionalities.

Figure 3-3 API Notification Call Flow



The above call flow represents an API notification request being processed by Converged Charging scenario in NEF:

1. PRODUCER_NF initiates a POST notification request to southbound NEF Ingress Gateway.
2. Southbound NEF ingress gateway sends this request to FiveGCAgent.
3. FiveGCAgent sends this request to NEF Service.
4. NEF Service sends this request to AF and also sends a response to PRODUCER_NF.
5. After completing the notification request, NEF processes the payload to create a CDR in FiveGCAgent. The actual notification message is with FiveGCAgent, which needs to be passed to CHF (asynchronous).
6. 5GC then creates the CDR message and forwards the same request to CHF.
7. NEF Service initiates a asynchronous converged post event charging request and charging request to CHF.
8. The asynchronous call reaches FiveGCAgent, which is then translated to a charging request and sent to southbound NEF Egress Gateway.
9. Southbound NEF Egress Gateway sends the request to CHF.
10. CHF processes the request and sends the response to NEF Service.

11. NEF creates CDR for all successfully processed as well as failed notifications. The `apiContent` parameter holds the JSON sent to AF for the notification.

Managing Converged Charging Support for NEF

Enable

You can enable the Converged Charging service by setting `qualityofservice.converged.pe.charging` parameter to true. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure this feature using Helm parameters. You can create the Model A, B, and D communication profiles using the `custom-values.yaml` file.

For information about configuring the `targetNfCommunicationProfileMapping.CHF`, `fivegcagent.chfBaseUrl`, `communicationProfiles`, and `egress-gateway` microservice parameters. See *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Observe

NEF provides the Traffic Influence related metrics for observing AF Session with this feature. For more information about the metrics, see [QoS Service Metrics](#).

3.3 Support for Device Trigger

The Device Trigger feature enables an Application Function (AF) to notify a particular User Equipment (UE) to perform application-specific tasks such as initiating communication with AFs, by sending a device trigger request through 5G core (5GC).

Device triggering is required when an AF does not contain information of IP address for the UE or if the UE is not reachable. Device trigger requests contain information required for an AF to send a message request to an appropriate UE and to route the message to the required application. This information processing method is known as trigger payload.

When an AF sends the device trigger message to an application at the UE side, the PDU Session establishment request is triggered. This is done by using the payload in the message, which contains information about the application at the UE side. After receiving this request, the application on the UE side initiates the PDU Session Establishment procedure.

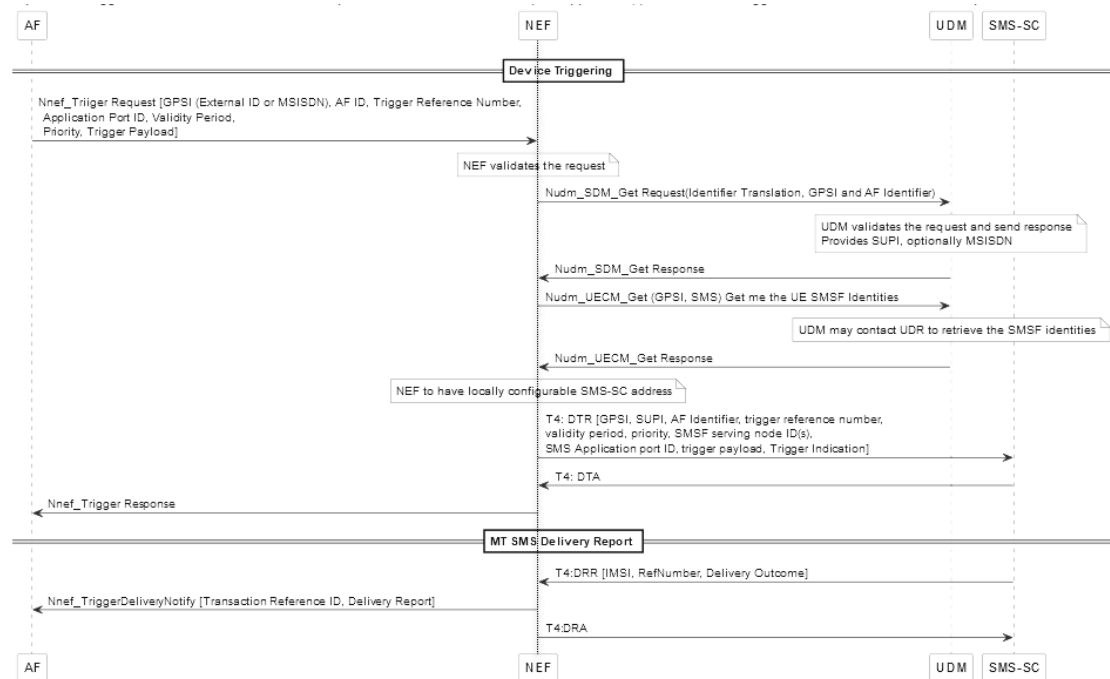
Device Trigger delivery status is notified by the SMS-SC to NEF in diameter message (delivery report request message). NEF then notifies the delivery status to the corresponding AF, which had created the Device Trigger transaction.

Note

This feature is required when AF is unable to establish communication with the UE.

Device Trigger Call flow

The following call flow scenario represents how an AF sends a Device Trigger (DT) request to NEF to notify a particular UE.

Figure 3-4 Device Trigger Call flow

The above call flow represents how an AF Device Trigger request is processed:

1. AF sends a `Nnef_Trigger` request, which includes either External ID or MSISDN of the UE to which the message must be sent, to NEF. This request consists of additional information such as Application Port ID, the Validity Period of the payload in the network, Priority, and the message content.
2. NEF validates this request by verifying if the AF is allowed to use the requested service. Then, it interacts with UDM to get the corresponding internal IDs and SMSF identities.
3. After receiving the required information from UDM, NEF sends a diameter request, which contains AF information, to SMS-SC.
4. When the diameter request is acknowledged with a DTA response to NEF, the request is sent to AF and the transaction is marked as complete.

Note

In case the UE is unavailable in the Device Trigger, the SMS-SC retains the data until the UE is available. When the UE is available, the payload is sent.

5. If payload is successful, it provides a DRR notification (either success or failure of message delivery) from SMS-SC to NEF.
6. Once NEF provides notification to AF, then it sends a DRA acknowledgment to SMS-SC.

Based on the reference number in the DRR, it can be identified for which Device Trigger request, the response is received.

Note

There can be failures due to validity being expired or UE being unreachable.

Managing the Device Trigger Feature**Enable**

You can enable the Device Trigger service by setting `global.enableFeature.deviceTrigger` parameter to true. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Note

The `global.enableFeature.convergedScefNef` parameter must be enabled in Helm configuration for this feature.

Configure

You can configure this feature using Helm parameters.

For information about configuring the `global.enableFeature.convergedScefNef`, `global.enableFeature.deviceTrigger`, `global.networkConfiguration.scsShortMessageEntity`, and `ocnef-diam-gateway.peerNodes` parameters, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Observe

NEF provides the Device Trigger related metrics and alerts for observing AF Session with this feature. For more information about the metrics, see [NEF Metrics](#). For more information about the alerts, see [Alerts](#).

3.4 API Invoker Onboarding and Offboarding

The API Invoker Onboarding and Offboarding feature enables Oracle Communications Cloud Native Core Network Exposure Function (NEF) to manage the exposure of the service APIs to different applications.

As per the 3GPP standard, API Invoker onboarding is the process of establishing trust between an API Invoker and an API Provider (NEF) to initiate a communication path. This process enables the API invoker application (external third-party applications or internal AF) to onboard the NEF system. Once the onboarding is complete, the invoker application is enabled to discover NEF and send service requests. The NEF services are provided based on the discovery group assigned to the invoker. For information related to discovery groups, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

As an API Provider, NEF uses the 3GPP defined CAPIF framework for onboarding and offboarding of API Invokers. The API Invoker Onboarding and Offboarding are defined as follows:

- **API Invoker Onboarding:** A one time process facilitated by CAPIF to enroll API invoker as a recognized application. The invoker negotiates the security methods and obtains authorizations from CAPIF for accessing the NEF services based on the discovery group

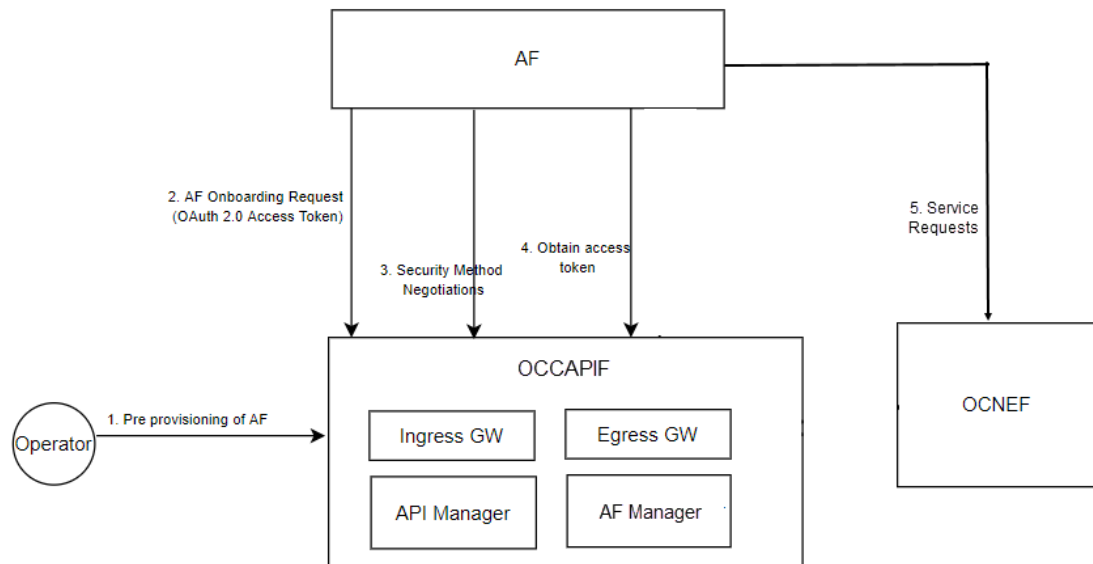
assigned to it. For more information about discovery group, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

- **API Invoker Offboarding:** The process to remove the API invoker as a recognized application of CAPIF. The API invoker triggers the offboarding procedure.

Example

The following diagram depicts an example of AF onboarding on CAPIF and establishment of communication with NEF.

Figure 3-5 Example of Communication Establishment between AF and NEF



The diagram provides the steps through which AF gets onboarded with CAPIF and then communicates with NEF:

1. **Pre provisioning of AF:** The operator sends a pre provisioning request to CAPIF. The request contains the key details of the AF along with the discovery group ID that needs to be assigned to the API invoker. For more information about discovery group, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.
As a response to this request, AF receives an access token to be included in the onboarding request (`onboardingSecret`) and the onboarding URI. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.
2. **Onboarding Request:** AF sends the `onboardedInvokers` POST request to CAPIF for onboarding. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.
3. **Security Method Negotiation:** NEF supports OAuth for service API invocation. Therefore, AF sends the `capif-security` PUT request to CAPIF. The API returns **OAuth** as the security method, in response. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.
4. **Obtain Authorization:** The AF sends a `capif-security` POST request to CAPIF.

The API returns the authorization code. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Note

The Authorization code must be included in each service request sent from AF to NEF through CAPIF.

For more information about the onboarding procedure, see the "Security procedures for API invoker onboarding" section in *3GPP Technical Specification 33.122, Release 16*.

5. After the successful onboarding, AF starts sending service requests to NEF.

Managing API Invoker Onboarding and Offboarding

Enable

API Invoker Onboarding and Offboarding is a core functionality of NEF. It remains enabled by default.

Observe

CAPIF provides the metrics for observing API Invoker Onboarding and Offboarding feature. For more information about the metrics, see [CAPIF Metrics](#).

3.5 Support for Model D Communication

As per the 3GPP 23.501 specifications, Network Functions (NFs) and services use different communication models to interact with each other. As a consumer NF, NEF communicates with different producer NFs, such as GMLC and UDM to handle the service requests coming from Application Functions (AFs). It supports the direct communication models and the Model D of the indirect communication method.

Support for Model D

In this model, NEF interacts with producer NFs through Service Communication Proxy (SCP).

The Model D Support functionality enables NEF to configure the necessary discovery and selection parameters required to find a suitable producer NF for handling a service request through SCP. SCP uses the discovery or selection parameters available in the request message to route the request to a suitable producer instance.

Model D Headers

NEF supports the following Model D headers that indicates the suitable target NF producer instance for NF service instance selection, reselection, and routing of subsequent requests:

Table 3-3 Supported Model D Headers

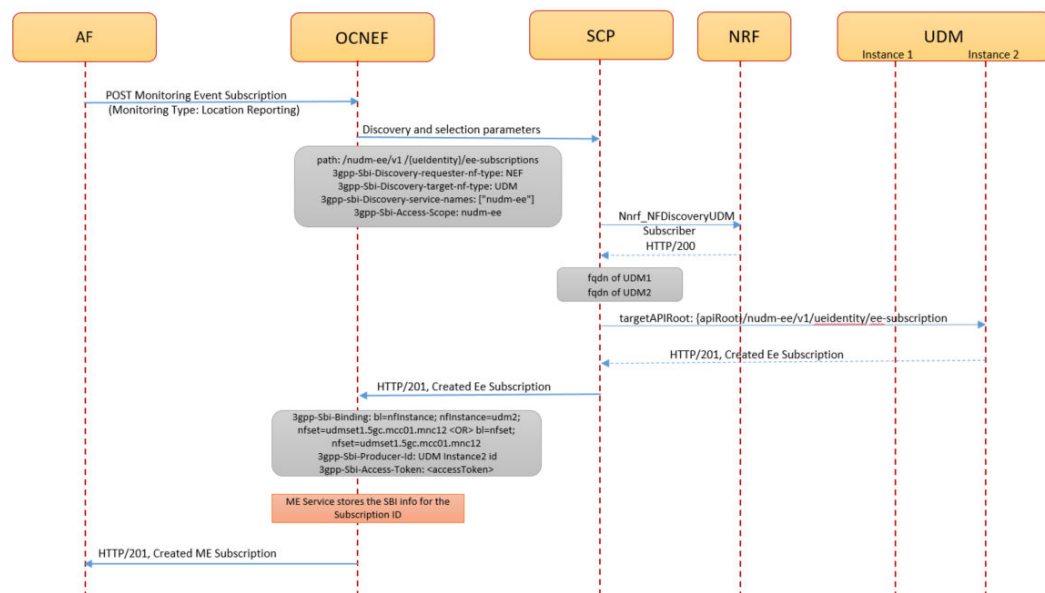
Header Name	Description
3gpp-Sbi-Target-Apiroot	<p>This header enables routing through SCP. The header contains the apiRoot of the target URI in a request sent to SCP during indirect communication.</p> <p>The header is used to indicate the apiRoot of the target URI when communicating indirectly with the HTTP server through SCP. This header is also used by SCP to indicate the apiRoot of the target URI if a new HTTP server is selected or reselected and there is no location header included in the response.</p> <p>Note: NEF does not include the <i>3gpp-Sbi-Target-apiRoot</i> header in the initial requests as it does not discover a producer NF. In subsequent requests, the header value is set to the apiRoot received earlier in the location header of the service responses from the NF Service Producer.</p>
3gpp-Sbi-Binding	<p>This header is used to communicate the binding information from an HTTP server for storage and subsequent use by an HTTP client.</p> <p>This header contains a comma-delimited list of Binding Indications from an HTTP server for storage and use of HTTP clients. The absence of this parameter in a Binding Indication in a service request is interpreted as "callback".</p>
3gpp-Sbi-Routing-Binding	<p>This header is used in a service request to send the binding information (3gpp-Sbi-Binding header) to direct the service request to an HTTP server, which has the targeted NF Service Resource context.</p> <p>This header enables alternate routing for subsequent requests at SCP. It contains a routing binding Indication to direct a service request to an HTTP server, which has the targeted NF service resource context.</p>
3gpp-Sbi-Producer-Id	This header is used in a service response from SCP to NEF. As a consumer NF, NEF uses this header to identify the producer NF.
3gpp-Sbi-Target-Nf-Id	The identity of the target NF that is being discovered.
3gpp-Sbi-Discovery-target-nf-type	The header indicating the type of the consumer NF
3gpp-Sbi-Discovery-requester-nf-type	The NF type of the Requester NF.
3gpp-Sbi-Discovery-service-names	<p>This header contains the service names used by the discovery NF. For example:</p> <p>when the target NF type is UDM, the 3gpp-Sbi-Discovery-service-names value must be: nudm-ee</p> <p>when the target NF type is GMLC, the 3gpp-Sbi-Discovery-service-names value must be: ngmlc-loc</p>

Table 3-3 (Cont.) Supported Model D Headers

Header Name	Description
3gpp-Sbi-Discovery-supported-features	List of features required to be supported by the target NF. This IE may be present only if the service-names attribute is present and if it contains a single service-name. It shall be ignored by the NRF otherwise.
3gpp-Sbi-Access-Scope	This header indicates the access scope of the service request for NF service access authorization.
3gpp-Sbi-Discovery-preferred-locality	This header indicates the preferred target NF location. For example, geographic location and data center.

Example

The following call flow describes an example of indirect communication between NEF and UDM using Model D functionality to cater to an ME Subscription request from AF:

Figure 3-6 Call Flow for ME subscription creation request through Model D indirect communication between NEF and UDM

The indirect communication between NEF and UDM is performed as follows:

1. To subscribe to the location monitoring service, AF sends a 3gpp-monitoring-event POST request to NEF.
2. The request routes through the external Ingress Gateway to CAPIF. After successful authorization, the CAPIF sends the 3gpp-monitoring-event POST request to the ME service with the AF identity (afid).

3. The ME service validates the request, creates a record in the database, generates a subscription, and sends the subscription details to the NEF 5GC Agent service. The 5GC Agent service processes and sends the following discovery and selection parameters to SCP through the 5GC Egress GW service:
 - a. parameters received through the NEF configuration
 - b. (only for subsequent messages) parameters received through the binding headers
4. SCP performs the NF discovery through NRF and receives the FQDN for the available UDM instances.
5. SCP sends the `Nudm_EventExposure` POST request to UDM. The request contains the UE identity for which location reporting is to be enabled.
6. On successful processing, UDM sends the `HTTP/201 <CreatedEeSubscription>` response to the SCP with the subscription details.
7. The SCP processes and sends the following information to ME Service through the 5GC Egress GW service:
 - a. `<CreatedEeSubscription>` response
 - b. The binding headers containing the UDM instances details for subsequent requests
8. ME service updates the NEF database with the following records:
 - a. The updated subscription records
 - b. The SBI information corresponding to the subscription ID. This information is used while routing the subsequent requests to the same UDM.
9. ME service sends the `201/HTTP <MonitoringEventSubscriptionCreated>` response to the AF.

Managing Support for Model D Communication

Enable

To enable the Model D functionality, the following configurations must be done using the custom value file for NEF:

- Enable SBI routing by setting the values of all the flags in `routesConfig.metadata` to **true** and configuring the `sbiRouting filterName1` under `routesConfig`. For more information, see [SbiRouting Configuration](#).
- Create NF Communication profile for Model D communication. For more information, see [Communication Profile Configuration](#).
- Assign the communication profile created for Model D communication to a target NF, such as UDM or GMLC. For more information, see [Target NF Communication Profile Mapping](#).

Configure

NEF allows you to configure NF Communication profiles and SCP for facilitating Model D functionality. The communication profiles can be mapped with target NFs to enable the Model D communication between NEF and the target NFs using the custom value file.

- **Configure SCP**

To integrate NEF with the SCP for Model D communication, perform the `sbiRouting`, `routesConfig`, `sbiRoutingErrorCriteriaSets`, and `sbiRoutingErrorActionSets` configurations for the Egress GW section of the custom values file for NEF.

Table 3-4 SCP Configuration

Parameter	Description
sbiRouting.sbiRoutingDefaultScheme	The value specified in this field is considered when 3gpp-sbi-target-apiroot header is missing. The default value is <code>http</code> .
peerConfiguration	Configurations for the list of peers. Each peer must contain the following: <ul style="list-style-type: none"> id host port apiPrefix
peerConfiguration.peerSetConfiguration	Configurations for the list of peer sets. Each peer set must contain the following: <ul style="list-style-type: none"> id httpConfiguration httpsConfiguration apiPrefix Each peer set must contain HTTP or HTTPS instances where in each instance contains priority and peer identifier, which maps to peers configured under <code>peerConfiguration</code> . No two instances should have same priority for a given HTTP or HTTPS configuration. In addition, more than one virtual FQDN should not be configured for a given HTTP or HTTPS configuration.
routesConfig.id	Specifies the ID of the route.
routesConfig.uri	Provide any dummy url, or leave the existing url with existing value
routesConfig.path	Specifies the path to be matched
routesConfig.order	Specifies the order of the execution of this route.
routesConfig.metadata.httpRuriOnly	Flag to enable httpRuriOnly functionality. When value is set to true, the RURI scheme is changed to http. For the value given as false, no changes are made to the scheme.
routesConfig.metadata.httpsTargetOnly	Flag to enable httpsTargetOnly functionality. When the value is set to true, SBI instances are selected for HTTPS list only (if 3gpp sbi-target root header is http). When the value is set to false, no changes are made to the existing scheme.
routesConfig.metadata.sbiRoutingEnabled	Flag to enable the sbiRouting for the selected route.
routesConfig.filterName1.name	Provide name as <code>SBIRouting</code> .
routesConfig.filterName1.args.peerSetIdentifier	Specifies the ID of the peerSetConfiguration.
routesConfig.filterName1.args.customPeerSelectorEnabled	
routesConfig.filterName1.args.errorHandling	The errorHandling section contains an array of errorCriteriaSet and actionSet mapping with priority. The errorCriteriaSet and actionSet are configured through Helm using <code>sbiRoutingErrorCriteriaSets</code> and <code>sbiRoutingErrorActionSets</code> . Note: To disable the rerouting under SBIRouting, delete the errorHandling configurations under <code>routesConfig</code> .

Table 3-4 (Cont.) SCP Configuration

Parameter	Description
sbiRoutingErrorCriteriaSets	The <code>sbiRoutingErrorCriteriaSets</code> contains an array of <code>errorCriteriaSet</code> , where each <code>errorCriteriaSet</code> depicts an ID, set of HTTP Methods, set of HTTP Response status codes, set of exceptions with <code>headerMatching</code> functionality.
sbiRoutingErrorActionSets	The <code>sbiRoutingErrorActionSets</code> contains an array of <code>actionset</code> , where each depicts an ID, action to be performed (Currently on REROUTE action is supported) and blocklist configurations.

Note

Note: The Model D functionality is based on SBI Routing. To enable SBI routing, the values of all the flags in `routesConfig.metadata` must be set to **true** and `sbiRouting.filterName1` must be configured.

- Configure Communication Profiles**

To configure the Model D communication profiles, perform the **communicationProfiles** configurations for the 5GCAgent section of the custom values file for NEF.

Table 3-5 Communication Profiles Configuration

Parameter	Description
<code>customModelD.discoveryHeaderParams.targetNfType</code>	The target NF, with which NEF is going to have the indirect communication. This parameter is mapped with the <code>3gpp-Sbi-Discovery-target-nf-type</code> discovery header. The parameter remains available only in the subsequent requests.
<code>customModelD.discoveryHeaderParams.discoveryServices</code>	The service names for the discovery NF. This parameter is mapped with the <code>3gpp-Sbi-Discovery-service-names</code> header.
<code>customModelD.discoveryHeaderParams.supportedFeatures</code>	This parameter is mapped with the <code>3gpp-Sbi-Discovery-supported-features</code> header
<code>customModelD.sendDiscoverHeaderInitMsg</code>	Indicates if the discovery headers must be sent in initial messages.
<code>customModelD.sendDiscoverHeaderSubMsg</code>	Indicates if the discovery headers must be sent in subsequent messages.
<code>customModelD.sendRoutingBindingHeader</code>	Indicates if the routing binding header must be included or not
<code>customModelD.discoveryHeaderParams.preferredLocality</code>	Preferred target NF location for the discovery NF. This parameter is mapped with <code>3gpp-Sbi-Discovery-preferred-locality</code> .

- Map Communication Profiles**

To enable the Model D communication with specific NF, assign the Model D profiles created in above configurations to the required NF. The following Helm parameters allow you to map the target NF type with the active profiles:

Table 3-6 Target NF Communication Profile Mapping

Parameter	Description
targetNfCommunicationProfileMapping. .UDM	The supported communication method for UDM Possible values are: <ul style="list-style-type: none"> — model A — model B — The communication profile configured for Model D in the communicationProfiles configuration. For more details about configuring Model D profiles, see Configure.
targetNfCommunicationProfileMapping. .GMLC	The supported communication method for GMLC Possible values are: <ul style="list-style-type: none"> — model A — model B — The communication profile configured for Model D in the communicationProfiles configuration. For more details about configuring Model D profiles, see Configure.

For more details related to configurations, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Observe

NEF provides the following NEF metrics for observing the Model D support functionality:

- ocnef_5gc_agent_srv_req_total
- ocnef_5gc_agent_srv_resp_total
- ocnef_5gc_agent_srv_latency_seconds
- ocnef_translation_count
- ocnef_translation_failure_count

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

3.6 Support for Security Token Generation

NEF supports the OAuth security method for service API invocation. This security procedure ensures that only authenticated service requests are directed to the core microservices for processing the request.

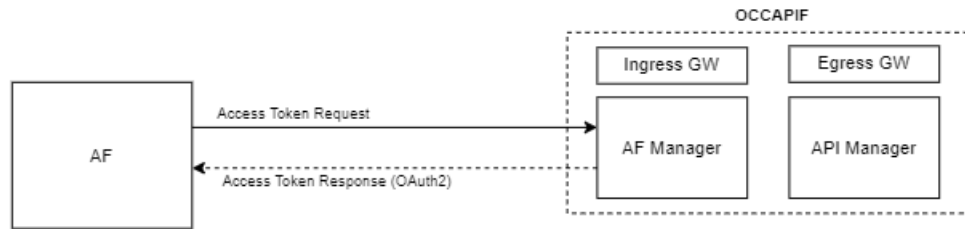
To achieve the secured interactions between AFs and NEF, the AF sends a request to the CAPIF to obtain authorization. The AF Manager service of the CAPIF generates an OAuth token (OAuth 2) and sends it to the AF to provide authorization.

① Note

NEF supports only OAUTH security method for service API invocation. There is no direct interface between AF and the NEF, hence, all the security procedures are handled by the CAPIF. The functionality returns only OAUTH security method by invoking a PUT operation.

The following diagram depicts an example of OAuth token generation by AF Manager service of CAPIF:

Figure 3-7 Security Token Generation

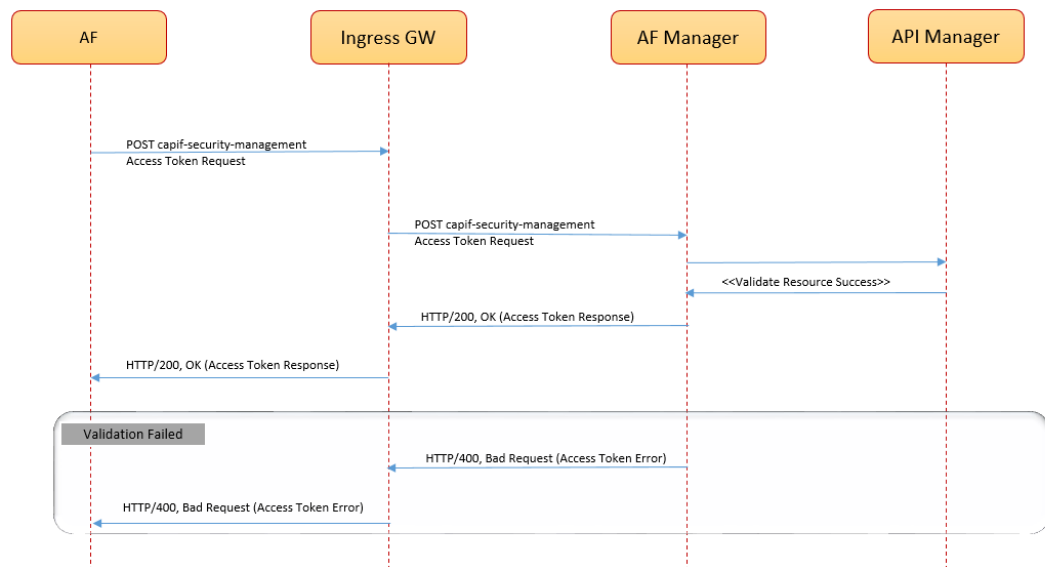


The API Invoker or AF attaches the received OAuth token with every service request sent to NEF. The OAuth token gets authenticated by the API Router service before the processing of the request at NEF.

The AF obtains authorization by sending a POST request to CAPIF.

The following diagram shows a call flow where NEF receives an authentication token request from AF, and the AF Manager service generates the token to provide authorization:

Figure 3-8 Call Flow for Security Token Generation by AF Manager



Note

The registration ID or API Exposing Function (AEF) ID changes during uninstallation of NEF. In such scenario, the manual deletion of security context and creation of new security context is necessary in order to generate new OAuth token. The AEF ID for security context can be obtained from NEF database.

For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Managing Security Token Generation

Enable

Security Token Generation is a core functionality of NEF. It remains enabled by default.

Observe

NEF provides the ME service related metrics for observing security functionality. For more information about the metrics, see [CAPIF Metrics](#).

3.7 Monitoring Event Service in NEF

In the network deployments, operators must track the events of User Equipment (UE) to provide customized and enhanced network services. Any change in the UE location is considered as an event and NEF facilitates third-party applications or internal Application Functions (AFs) to monitor and get the report about such events.

The ME service monitors the UE specific events, such as change in UE location, details of network area, PDU status, connectivity loss, and UE reachability, then provides such information. NEF communicates with the following network functions to process the monitoring event requests:

- Unified Data Management (UDM)
- Access and Mobility Management Function (AMF)
- Session Management Function (SMF)

Depending on the specific monitoring event or information, UDM and AMF collects the information and sends it through NEF.

Monitoring Event Service

The purpose of the Monitoring Event feature is to provide the following information:

- Current location of UE
- Last known location of UE with time stamp
- PDU Session status
- Loss of Connectivity Event
- UE Reachability Event

This functionality is achieved by using the 3GPP defined monitoring event based on the monitoring type. The event is detected based on the event reporting parameters received in the monitoring event subscription request as follows:

- One-time reporting
- Maximum number of reports
- Maximum duration of reporting
- Monitoring type

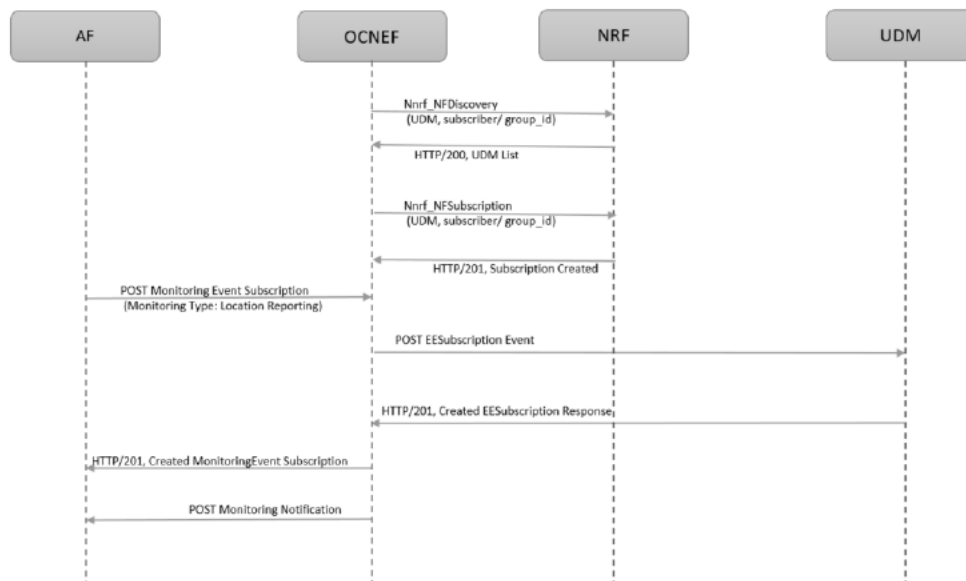
Note

- NEF scope for handling `externalGroupId` and `maximumNumberOfReports`:
 - If a UE is part of multiple `externalGroupId` across multiple subscriptions, then NEF does not aggregate and sends one report for each UE. Instead, NEF forwards all the reports received from SMF to AF.
 - NEF tracks only the number of UEs in a group and not the individual UE. NEF accepts a total of either (`maximumNumberOfReports` * `numberOfUEs`) cumulatively or until the subscription expiration based on the `monitorExpireTime` parameter, whichever is earlier.
 - NEF does not track the number of reports at individual UE level in case of `externalGroupId` subscription.
- NEF scope for handling `externalGroupId` and `groupReportGuardTime`:
 - For `PDN_CONNECTIVITY_STATUS`, NEF does not aggregate the reports based on the guard time. SMFs aggregate the reports and send them to NEF. Then, NEF forwards the same report to AF.
 - For `UE_REACHABILITY` and `LOSS_OF_CONNECTIVITY`, NEF does not aggregate the reports based on the guard time. AMF forwards the request for each UE in the group and the same is forwarded to AF. There is no aggregation of reports occurs at NEF.

For more information about ME parameter, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

To provide the monitoring service to AF, NEF interacts with UDM by using the `Nudm_EventExposure` service. For more information about this service, see [3GPP Technical Specification, Unified Data Management Services, Release 17](#).

The following diagram represents a high-level call flow where NEF receives a location tracking request from AF and interacts with UDM to obtain the required information:

Figure 3-9 Call Flow for Monitoring Event Subscription

The above call flow can be described as follows:

1. The NRF Client communicates with NRF for UDM Discovery (`Nnrf_NFDiscovery`) and receives a list of available UDMs with their `Subscriber IDs` or `Group IDs`.

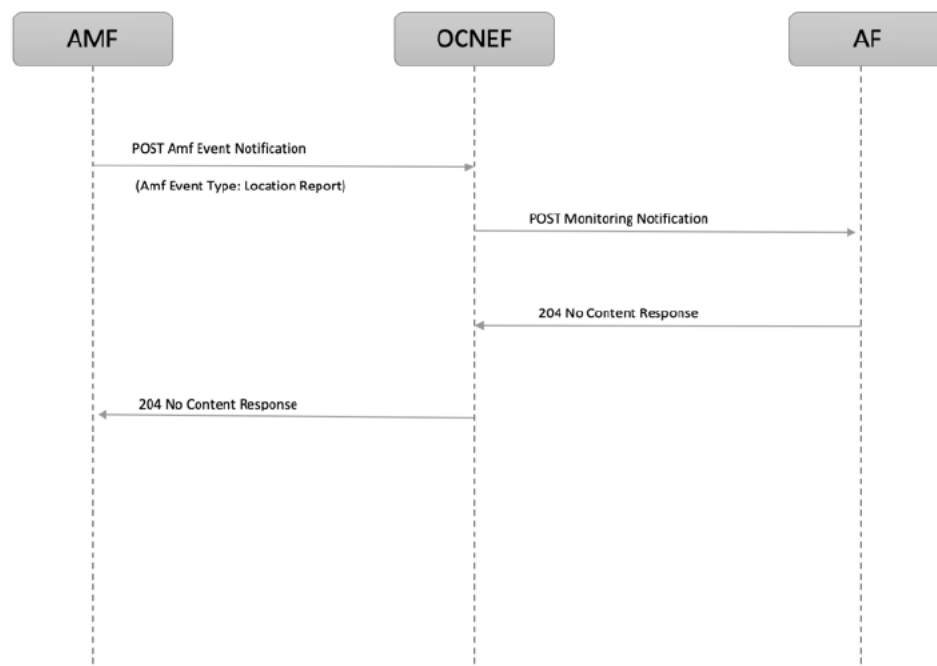
Note

The UDM discovery is performed even before NEF starts receiving any subscription requests from AF.

2. AF sends a POST request (`3gpp-monitoring-event`) for monitoring event subscription to NEF.
3. After the authentication of the request through CAPIF, the request reaches the ME service.
4. The ME service processes the request and sends a POST request (`Nudm_EventExposure`) for event notification subscription to the selected UDM from the list available through UDM discovery.
NEF subscribes to this event to receive the monitoring event of a UE, and Updated Location of the UE when UDM becomes aware of a location change of the UE. For more information about `Nudm_EventExposure`, see [3GPP Technical Specification, Release 16, Access and Mobility Management Services](#).

After successful subscription with the UDM, NEF starts receiving the event reports from AMF and sends it to AF.

The following diagram represents a high-level call flow where NEF receives a event report from AMF and forwards it to AF:

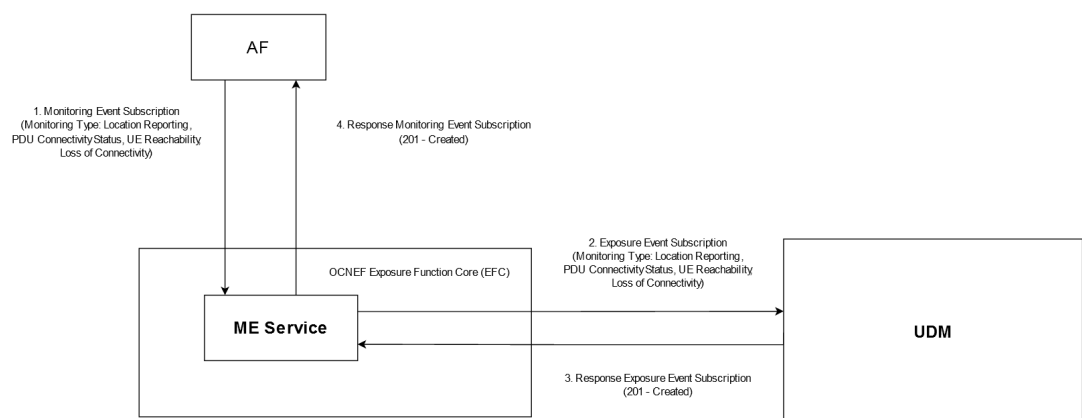
Figure 3-10 Call Flow for Monitoring Event Reporting

The call flow can be described as follows:

1. AMF sends a POST request (**AMF-event-notification**) for monitoring event notification to NEF.
2. After the processing of the request NEF sends the event report to AF.

ME Subscription Creation

The following call flow is an example of how NEF receives a subscription request for location tracking from AF and interacts with UDM to get the relevant information:

Figure 3-11 Example of Monitoring Event Subscription for Creation

The above mentioned call flow provides the steps to process the service request. These steps can be described as follows:

1. After successful registration and onboarding, the AF sends a POST request (3gpp-monitoring-event) to create a subscription for the supported monitoring event toward NEF.
2. When the request is authenticated by NEF Exposure Gateway (EG), the ME service processes the request based on the type of the event. For any monitoring type event, NEF sends a POST request (Nudm_EventExposure) to create a subscription to UDM.

NEF subscribes to the event in order to receive the notification for the event based on the monitoring type. For more details related to the `Nudm_EventExposure` service, see [3GPP Technical Specification, Unified Data Management Services, Release 17](#).

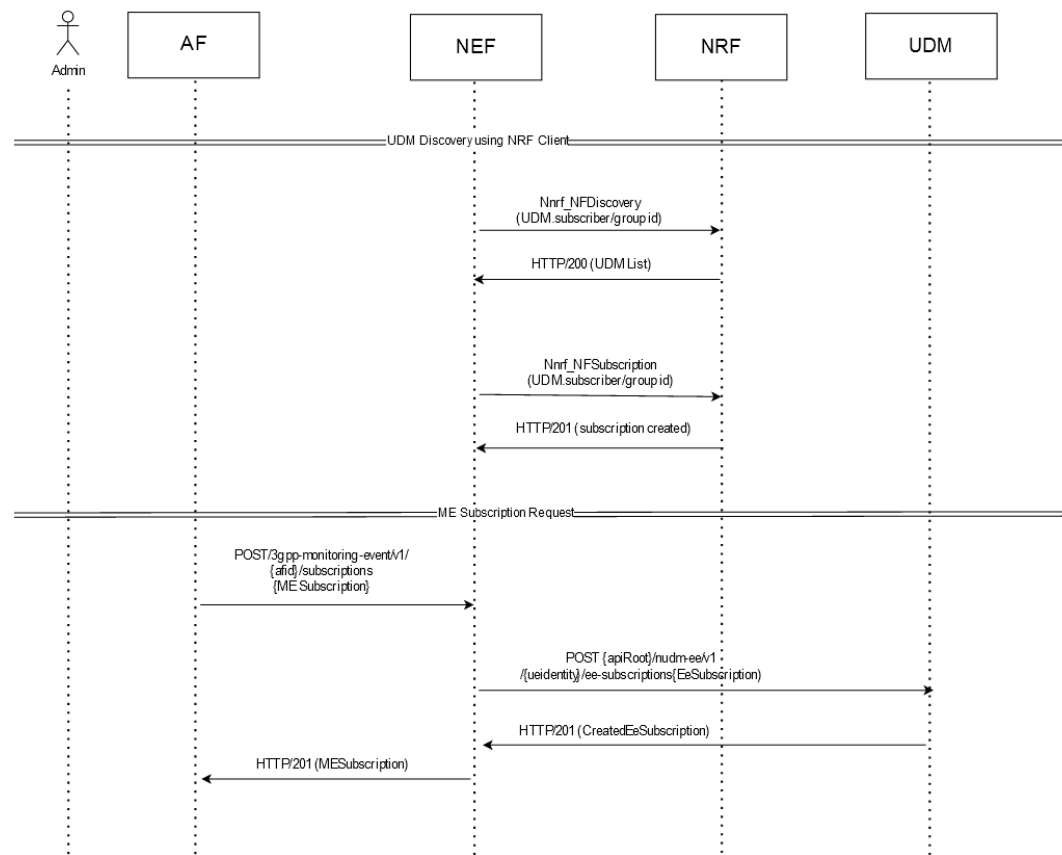
3. On successful processing of the request, UDM sends an acknowledgment for subscription creation. In case of failure, the error response is sent.
4. After receiving the successful response from the UDM, the ME service sends the subscription creation acknowledgment response to the AF. This creates the ME service subscription.

Note

The expired ME subscriptions get deleted by the NEF Audit service.

Create ME Subscription

The following diagram depicts a sample call flow for ME subscription creation by AF for location tracking:

Figure 3-12 Example of ME Subscription Call Flow

The different types of requests shown in the call flow are described as follows:

1. By default, the NRF Client communicates with NRF for UDM Discovery (`Nnrf_NFDIScovery`) and receives a list of available UDMs with their Subscriber IDs or Group IDs.

Note

The UDM discovery is performed even before NEF starts receiving any subscription requests from AF.

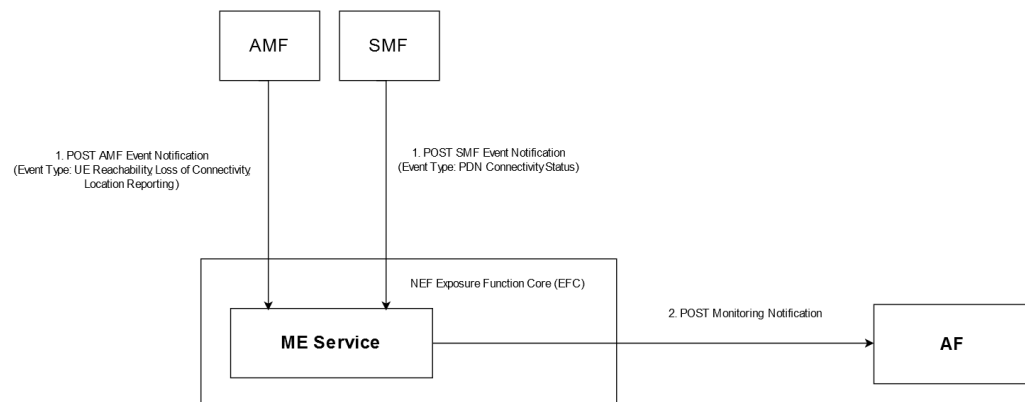
2. To subscribe to ME service, AF sends a `3gpp-monitoring-event` POST request to NEF.
3. NEF performs the following steps:
 - a. The request routes through the external Ingress Gateway to NEF Exposure Gateway (EG). The API Router performs the authentication process and checks if the AF is authorized to access the requested service. Based on the authentication result, one of the following actions is performed:
 - If the authorization fails, then API Router sends the `HTTP/403 <Unauthorised>` response to AF and closes the request.
 - If the authorization is successful, then API Router sends the `HTTP/200 <Authorized>` response to the API Manager.

- b. After successful authorization, the API Router service sends the `3gpp-monitoring-event` POST request to the ME service with the AF identity (`afid`).
 - c. The ME Service validates the request, creates a record in database, generates a subscription, and sends the subscription details to the 5GC Agent service.
 - d. The 5GC Agent service processes the subscription information, interacts with the Config-Server service, and receives the list of UDMs available through NRF discovery.
 - e. After successful UDM discovery, 5GC Agent sends the `Nudm_EventExposure` POST request to UDM through the 5GC Egress Gateway. The request contains the UE identity for which location reporting can be enabled.
4. After the processing is complete, UDM sends the HTTP/200 `<CreatedEeSubscription>` response to the NEF with the subscription details.
 5. The 5GC Agent service in NEF routes the HTTP/200 `<CreatedEeSubscription>` response to ME Service. The service updates the NEF database with the updated subscription records and sends the 200/HTTP `<MonitoringEventSubscriptionCreated>` response to the AF through the external Ingress Gateway.
 6. This creates the monitoring event subscription in NEF for the AF to monitor the location of the specified UE.
The subscription remains valid for the number of days defined in the POST request through the `monitorExpireTime` parameter.

Sending Notifications to AF

The following diagram depicts an example, where NEF receives the event reports from AMF and sends it to the AF:

Figure 3-13 Example of Sending Event Reports through ME Notification



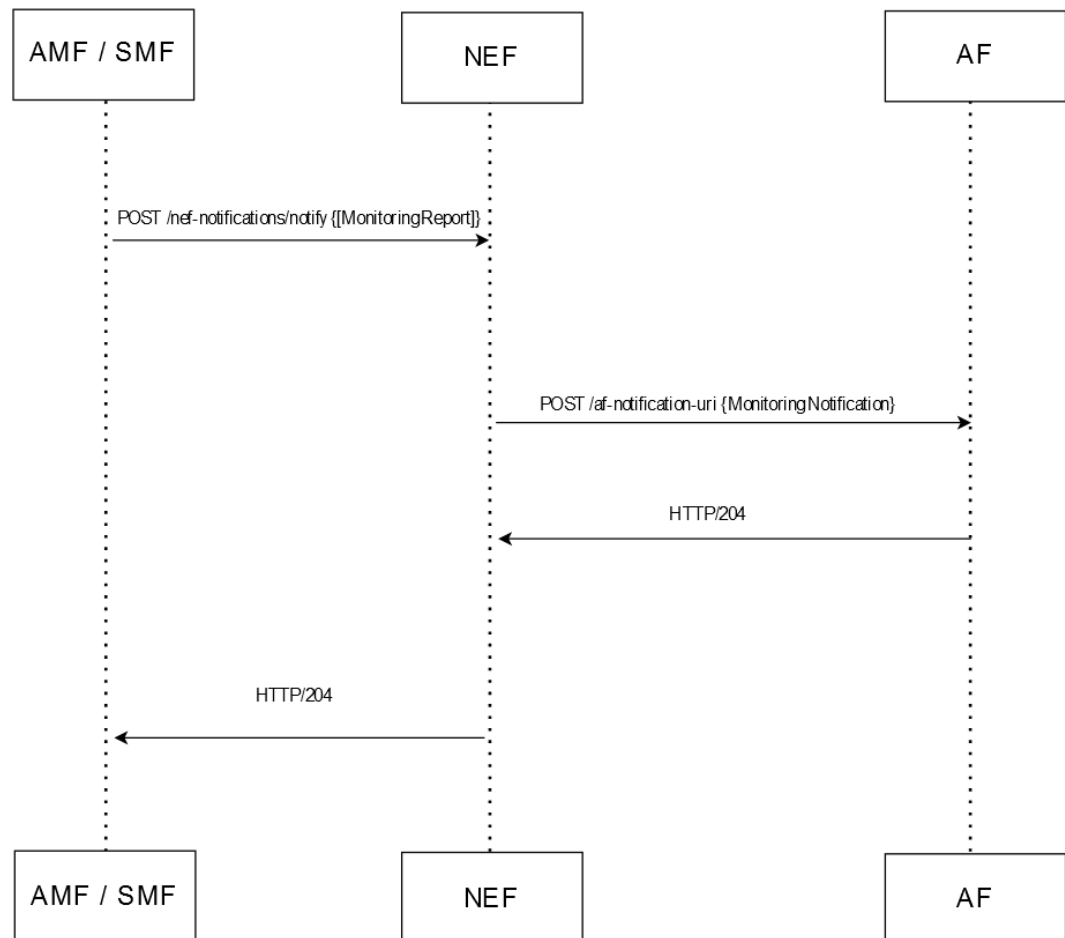
The diagram provides the steps through which the ME Notification or reports are sent to AF. These steps are described as follows:

1. After successful creation of the ME subscription, the AMF or SMF sends POST request (`AMF-event-notification`) for creating a monitoring event notification to NEF.
2. After processing the request, NEF starts sending the response to AF.

Send ME Notification

The following diagram depicts the sample call flow for sending monitoring event reports as ME notifications from AMF or SMF to AF through NEF:

Figure 3-14 Example of ME Notification Call Flow



The different types of requests in the above mentioned call flow are described as follows:

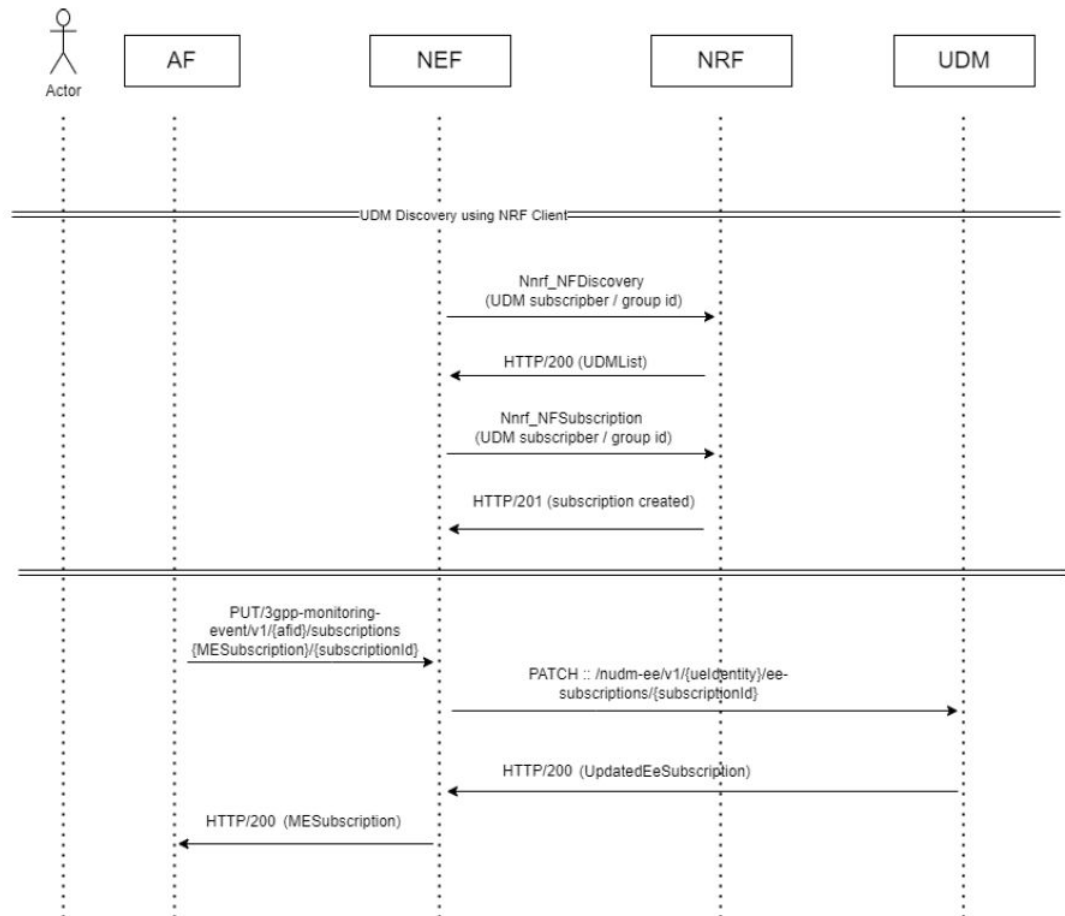
1. AMF or SMF sends a `nef-notifications/notify` POST request to NEF with the subscriber details corresponding to the ME subscription created in the above mentioned call flow.
2. The request routes through the 5GC Ingress Gateway and 5GC Agent services in NEF to ME Service.
3. The ME Service processes the request, creates a record in database, and sends the `ME Notification` POST request to the corresponding AF with the subscriber location details.
4. This sends the response of the specified UE to the AF.

For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Update ME Subscription

The following diagram depicts a sample call flow for updating a ME subscription for location tracking:

Figure 3-15 Example of Updating a ME Subscription - Call Flow



The different types of requests shown in the call flow are described as follows:

1. By default, the NRF Client communicates with NRF for UDM Discovery (Nnrf_NFDDiscovery) and receives a list of available UDMs with their Subscriber IDs or Group IDs.

Note

The UDM discovery is performed even before NEF starts receiving any subscription requests from AF.

2. To update an ME subscription, AF sends a 3gpp-monitoring-event PUT request to NEF.
3. NEF performs the following steps:
 - a. The request routes through the external Ingress Gateway to NEF Exposure Gateway (EG). The API Router performs the authentication process and checks if the AF is

authorized to access the requested service. Based on the authentication result, one of the following actions is performed:

- If the authorization fails, then API Router sends the HTTP/403 <Unauthorised> response to AF and closes the request.
 - If the authorization is successful, then API Router sends the HTTP/200 <Authorized> response to the API Manager.
- b. After successful authorization, the API Router service sends the 3gpp-monitoring-event PUT request to the ME service with the AF identity (afid).
 - c. The ME Service validates the request, updates the subscription, and sends the details to the 5GC Agent service.
 - d. The 5GC Agent service processes the subscription information, interacts with the Config-Server service, and receives the list of UDMs available through NRF discovery.
 - e. After successful UDM discovery, 5GC Agent sends the Nudm_EventExposure PATCH request to UDM through the 5GC Egress Gateway. The request contains the UE identity for which location reporting can be enabled.

Note

PUT functionality is supported from AF to NEF. PATCH functionality is supported from NEF to UDM. Hence, here PUT is converted to PATCH and then send through UDM.

4. After the processing is complete, UDM sends the HTTP/201 <UpdatedEeSubscription> response to the NEF with the subscription details.
5. The 5GC Agent service in NEF routes the HTTP/201 <UpdatedEeSubscription> response to ME Service. The service updates the NEF database with the updated subscription records and sends the 201/HTTP <MonitoringEventSubscriptionUpdated> response to the AF through the external Ingress Gateway.
6. This creates the monitoring event subscription in NEF for the AF to monitor the location of the specified UE.
The subscription remains valid for the number of days defined in the PATCH request through the `monitorExpireTime` parameter. For more information about this parameter, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Monitoring Event Reporting

Enable

Monitoring Events Reporting is a core functionality of NEF. It remains enabled by default.

Observe

NEF provides the ME service related metrics for ME Service feature. For more information about the metrics, see [NEF Metrics](#).

3.7.1 Converged SCEF NEF Model for Monitoring Event

In a Monitoring Event (ME) call flow in 5G network, when a User Equipment (UE) is latched to the 5G network and AF has subscribed for ME service, the AMF node detects an event and triggers notification to the Network Exposure Function (NEF) in the following sequence:

1. AF sends request to NEF.

2. NEF creates subscription on UDM.
3. UDM sends the send request to AMF.
4. On event detection, AMF sends notification to NEF.

If in case the UE is latched to a 4G network, then NEF cannot create subscription or receive notification from that network. NEF needs to be enabled with the Converged SCEF-NEF solution for specific services where interaction between the Service Capability Exposure Function (SCEF) and NEF is required.

When there is a UE capable of mobility between Evolved Packet Core (EPS) and 5G Core (5GC), the network is expected to associate the UE with an SCEF+NEF node for Service Capability Exposure.

With the introduction of this feature, when a UE request is sent from a 4G network, MME node detects this event and sends the notification to NEF. As NEF is not capable of accepting the diameter requests, a Diameter Gateway is used to allow the diameter traffic to NEF.

Note

- The remaining notification flow is similar to the existing flow from NEF to AF.
- The `convergedScefNefEnabled` parameter:
 - introduces `enableFeature.convergedScefNef` diameter gateway for receiving diameter traffic from EPC network nodes.
 - enables EPC subscription parameters for subscriptions towards UDM (currently `monitoringEvents` Location reporting subscriptions).

For example: MME configuration under diameter gateway.

```
ocnef-diam-gateway:
  clientPeers: |
    - name: 'mme'
      realm: 'mme.oracle.com'
      identity: 'T6-client'
```

where, MME is a client diameter node.

Note

- During upgrade, only the subscription request on the new pod creates subscription on HSS (epc network). This does not apply for Gateway Mobile Location Centre (GMLC).
If GMLC returns fail, based on the `switchToUdmOnFailure` parameter, NEF performs failover to UDM, then NEF sends the parameter to UDM to create subscription on Evolved Packet Core (EPC) when `converged_scef_nef` is enabled.
- SCEF-Reference-ID is the identifier for the notification from MME to find the corresponding subscription.
- If subscription creation at UDM is successful, but there was failure while creating subscription at HSS then the NEF considers the subscription creation to be successful. `MeEPCCAddSubscriptionFailureRateCrossedThreshold` alert is triggered when this failure happens at EPC.

Monitoring Events Call Flow in Converged SCEF-NEF

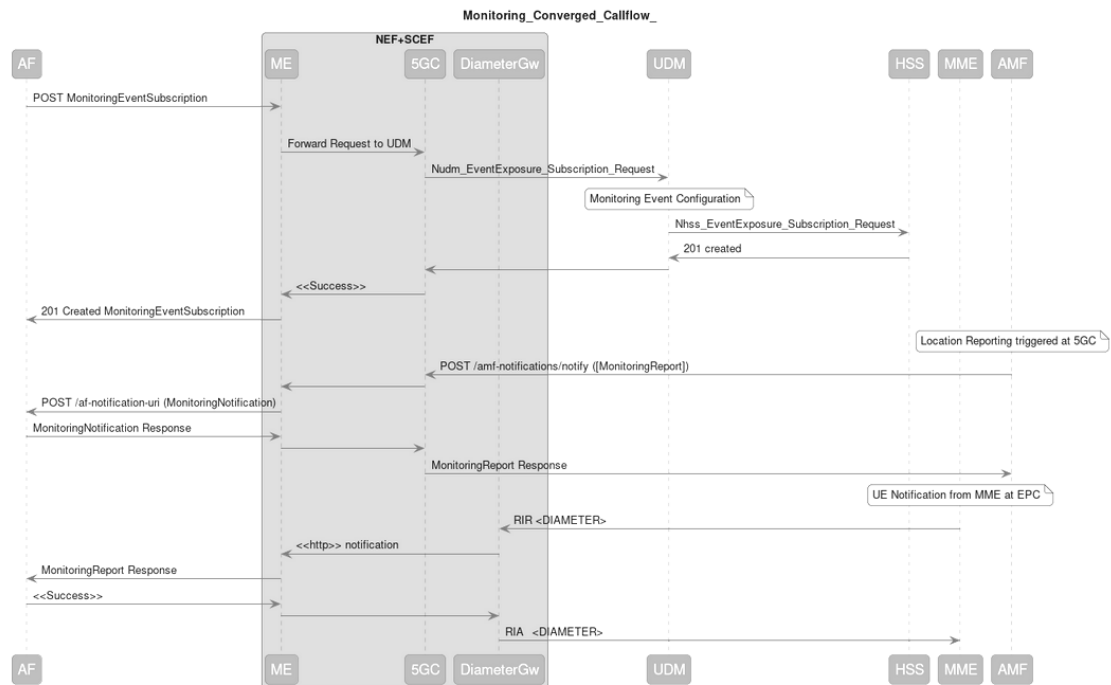
This section provides call flow supported for Converged SCEF-NEF functionality with separate UDM and HSS.

Generally, a UE request for ME subscription is handled as follows:

- If the UE request is from 5G network
 1. Application Function (AF) sends request to NEF for ME subscription.
 2. NEF sends EeSubscription request to UDM.
 3. UDM creates the subscription on AMF.
 4. AMF sends this event notification to NEF directly.
- If the UE request is from 4G network
 1. Application Function (AF) sends request to NEF for ME subscription.
 2. NEF sends EeSubscription request to UDM. If the `epcAppliedInd` parameter is set to True, the request also applies to 4G.
 3. UDM creates the subscription on HSS along with AMF.
 4. HSS creates subscription in MME.
 5. MME sends RIR message T6X interface event notification to NEF through DiameterGw.

In the following scenario, SCEF+NEF configures events at UDM, indicating that the event may occur in 5GC as well as EPC. Based on this information, UDM configures events at 5GC and also invokes event subscription at HSS, which in turn configures events at EPC.

SCEF+NEF receives event notifications from 5GC or UDM as per 5GC procedures, whereas EPC reports events through 4G procedures (diameter based messages).

Figure 3-16 Monitoring Events - SCEF-NEF Call Flow

The above call flow scenario represents a ME subscription request being processed when Converged SCEF-NEF feature is enabled:

1. AF sends a `MonitoringEventSubscription` request to ME.
2. ME forwards the request to 5GC.
3. 5GC sends subscription request to UDM.
4. UDM creates subscription on AMF and HSS. HSS sends information to MME.
5. If the event detected is at 5G network, then AMF sends this response notification to 5GC. If the event detected is at 4G network, then MME sends this response notification to NEF through DiameterGw. Diameter is translated to HTTP internally in NEF.
6. NEF sends the notification to AF.
7. AF sends a success response to NEF.
8. NEF sends this response directly to AMF and to MME through DiameterGw.

3.8 GMLC Based Location Monitoring

In the network deployments, there are certain scenarios when operators need highly accurate location details of subscribers that contains the geographic information. Such requests cannot be processed using the UDM or AMF based location monitoring. To cater to these type of requests, NEF supports the Gateway Mobile Location Center (GMLC) based location monitoring along with UDM and AMF based monitoring.

NEF selects the GMLC or UDM/AMF based location service as per the service requirements, such as location QoS, whether an immediate or deferred subscriber location is requested, and the availability of GMLC, UDM, or AMF.

GMLC Based Location Monitoring

The GMLC Based Location Monitoring feature enables NEF to monitor the current or last known location of subscribers or User Equipment (UE) based on the requests received from AFs. The feature caters to the following types of location monitoring requests:

- **Immediate Requests:** One time request for location monitoring of a subscriber
- **Deferred Requests:** Request to monitor the subscriber location for:
 - a specific number of location updates
 - specific duration

In such requests, NEF receives periodic updates about the subscriber location from GMLC and it further sends these details to the AFs.

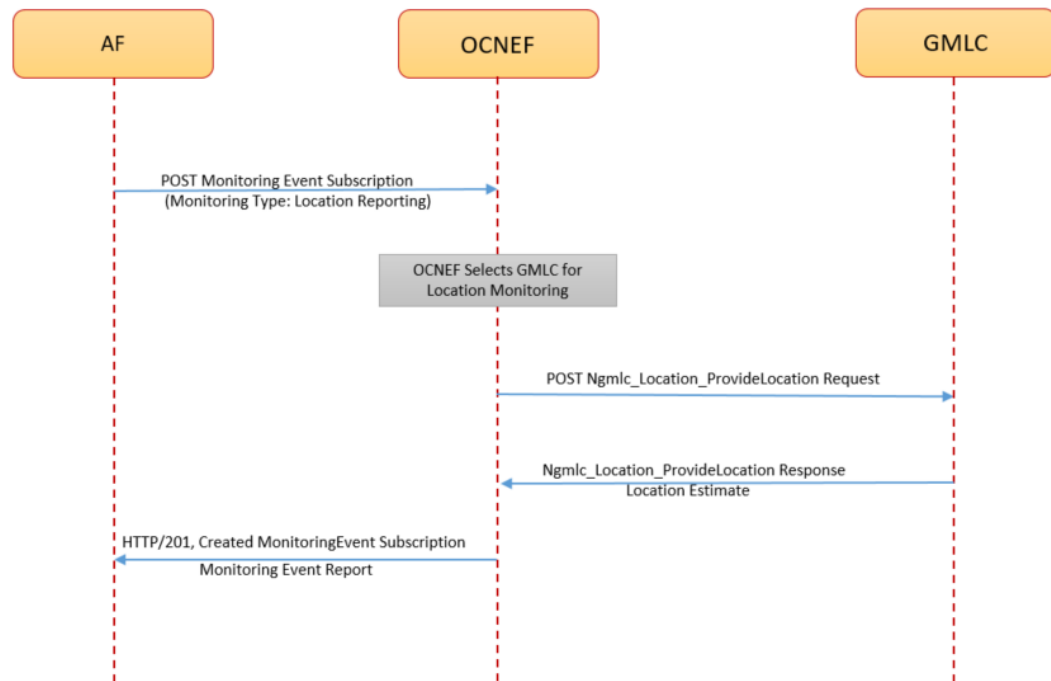
To setup GMLC based location monitoring, NEF invokes the Home GMLC that is responsible to control the privacy checking of the target subscriber. The invoked GMLC further interacts with the core NFs to provide the subscriber location details.

NEF exposes the location information as and when received from GMLC to the AF. The information is exposed based on the 3GPP specifications provided for the Northbound APIs towards the AF. For more information about the specifications, see 3GPP TS 29.522 Rel. 16.9.

Immediate Requests

The following diagram shows a high-level call flow where NEF receives a subscription request for the immediate location of a subscriber from AF. NEF processes the request, evaluates the location service provider as GMLC, and interacts with GMLC to get the information:

Figure 3-17 Call Flow for GMLC Based Monitoring Event Subscription for Immediate Request



The call flow can be described as follows:

1. AF sends a `3gpp-monitoring-event` POST request to NEF to subscribe to an immediate location monitoring service.

Note

For an immediate location monitoring request, the `MaximumNumberOfReports` parameter value remains **1** and `ldrType` must not be present. For more information about the parameters, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

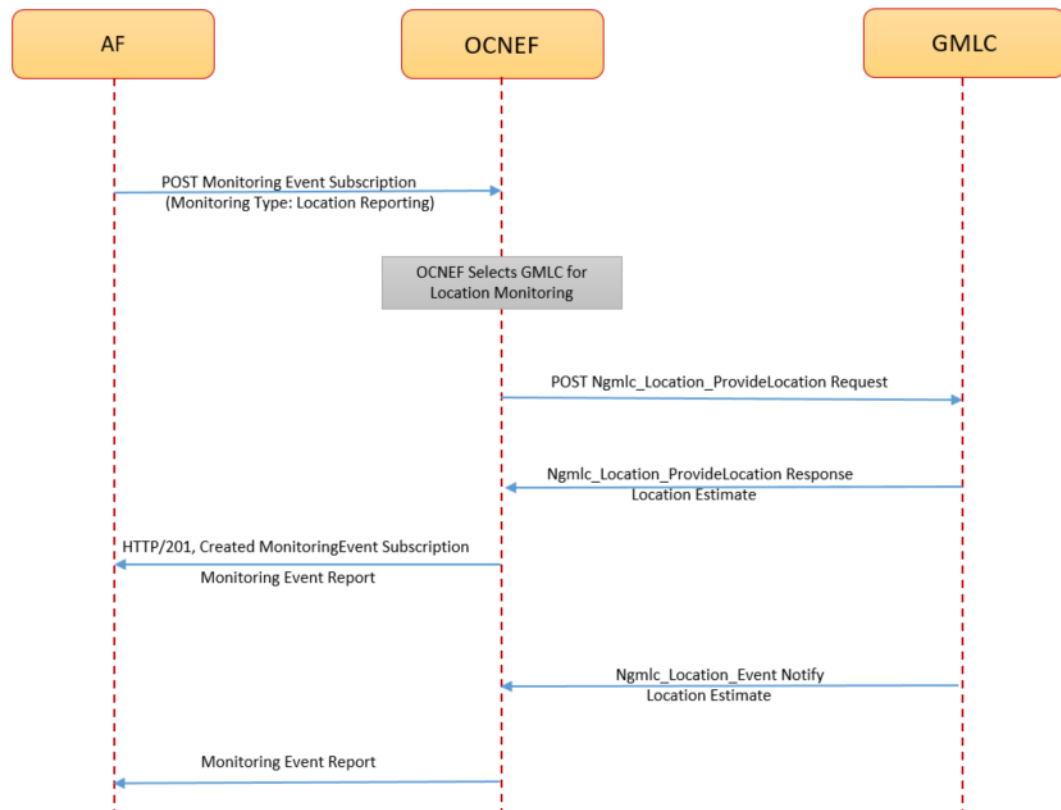
2. The request routes through the external Ingress Gateway to NEF ME service after a successful authentication process in NEF CAPIF.
3. The ME service performs the following tasks:
 - a. validates the request
 - b. evaluates the Location Provider as GMLC
For more information about how NEF determines the location provider, see [Evaluation of Location Provider](#).
 - c. generates a subscription
 - d. sends the subscription details to the 5GC Agent service.
4. The 5GC Agent service processes the subscription information, interacts with the Config-Server service, and performs the discovery of the GMLCs deployed within the network through NRF or SCP, based on the selected communication model.
5. After successful GMLC discovery, the 5GC Agent sends the `Ngmlc_Location_Provide_Location` POST request to GMLC through the 5GC Egress Gateway. The request contains the UE identity for which location reporting is to be enabled.
6. On successful processing, GMLC sends the `HTTP/200` response to the 5GC Agent service. The response contains the subscriber location information.
7. NEF translates and forwards the location information to AF without storing the data in the database.

Note

NEF clears the database before sending the location information to the AF.

Deferred Requests

The following diagram shows a high-level call flow where NEF receives a subscriber location information for a defined period or based on the geographic area of a subscriber from AF. NEF processes the request, evaluates the location service provider as GMLC, and interacts with GMLC to get the information:

Figure 3-18 Call Flow for GMLC Based Monitoring Event Subscription for Deferred Request

The call flow can be described as follows:

1. AF sends a 3gpp-monitoring-event POST request to NEF to subscribe to a deferred location monitoring service.

Note

For a deferred location monitoring request, the `ldrType` parameter must be present. For more information about the parameters, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

2. The request routes through the external Ingress Gateway to NEF ME service after a successful authentication process in CAPIF.
3. The ME service performs the following tasks:
 - a. validates the request
 - b. evaluates the Location Provider as GMLC
For more information about how NEF determines the location provider, see [Evaluation of Location Provider](#).
 - c. generates a subscription
 - d. sends the subscription details to the 5GC Agent service.

4. The 5GC Agent service processes the subscription information, interacts with the Config-Server service, and performs the discovery of the GMLCs deployed within the network through NRF or SCP, based on the selected communication model.
5. After successful GMLC discovery, the 5GC Agent sends the `Ngmlc_Location_Provide_Location` POST request to GMLC through the 5GC Egress Gateway. The request contains the UE identity for which location reporting is to be enabled.
6. On successful processing, GMLC sends the `HTTP/200` response to the 5GC Agent service. The response contains the subscriber location information.
7. NEF translates and forwards the location information to AF without storing the data in the database.

Note

NEF clears the database before sending the location information to the AF.

8. GMLC uses the `Ngmlc_Location_Event_Notify` service to send further subscriber location information to NEF as per the deferred request.

Evaluation of Location Provider

NEF selects GMLC based location monitoring as per the operator configurations and the presence of the `LocQoS` parameter in AF location monitoring request. The following table describes the logic used by NEF to select the GMLC based monitoring over the UDM/AMF based monitoring:

Table 3-7 Evaluation of Location Provider

Is GMLC Feature Enabled	If LocQoS parameter is present in the ME Subscription Request	Selected Location Monitoring Destination
No	Yes	UDM/AMF
No	No	UDM/AMF
Yes	Yes	GMLC
Yes	No	Based on operator configuration for <code>destIfLocQoSAbsent</code> parameter in Custom YAML file. For more details about the configurations, see Managing GMLC Based Location Monitoring .

On receiving the subscription request from AF, NEF verifies the configuration for the **gmlcDeployed** parameter in the helm configurations. For more details about the configurations, see [Managing GMLC Based Location Monitoring](#).

1. If the value for `gmlcEnabled` parameter is **false**, then the location tracking request is sent to UDM
2. If the value for `gmlcDeployed` parameter is **true**, then NEF checks if the subscription request contains the `locQoS` parameter. For more information about the request parameters, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.
 - a. If the request does not contain the `locQoS` parameter, NEF checks the configuration of the `destIfLocQoSAbsent` parameter in the HELM configurations:

- If the value of `destIfLocQosAbsent` is **UDM**, then the subscription request is sent to UDM
 - If the value of `destIfLocQosAbsent` is **GMLC**, then the subscription request is sent to GMLC
- b. If the request contains the `locQos` parameter, then NEF compares the values of `hAccuracy` and `vAccuracy` parameters of the subscription request with the `gmlchAccuracy` and `gmlcVaccuracy` parameters configured in HELM chart.
- If the values of subscription request parameters are same or greater than the HELM configuration values, then the request is forwarded to GMLC.
 - If the values of subscription request parameters are smaller than the HELM configuration values, then the request is forwarded to UDM.

For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Managing GMLC Based Location Monitoring

Enable

The GMLC Based Location Monitoring functionality can be enabled during NEF deployment using the custom `value.yaml` file.

To enable this feature, set the `gmlcEnabled` parameter to **true** under the global configurations in the custom-values.yaml file for NEF.

The **gmlcEnabled** parameter values can have the following values:

- **true**: The value states that GMLC is deployed in the current deployment and GMLC based location monitoring can be considered when NEF receives a location tracking request from AF
- **false**: The GMLC based location monitoring is not considered when NEF receives a location tracking request from AF.

Configure

The GMLC Based Location Monitoring functionality can be configured during NEF deployment using the custom-values.yaml file. The following parameters must be updated in the custom values file for NEF:

Table 3-8 GMLC Based Location Monitoring Parameters

Parameter	Description
<code>monitoringevents.gmlc.destIfLocQosAbsent</code>	Indicates the location provider when <code>LocQos</code> is not present in the request. For more details about the <code>LocQos</code> parameter, see <i>Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide</i> .
<code>monitoringevents.gmlc.switchToUdmOnFailure</code>	Specifies if the location reporting event request must be sent to UDM when NEF receives failure from GMLC.
<code>monitoringevents.gmlc.switchOnErrorCodes</code>	Contains a list of error codes, based on which NEF switches to UDM.

Table 3-8 (Cont.) GMLC Based Location Monitoring Parameters

Parameter	Description
monitoringevents.gmlc.explicitCancellation	Indicates whether an explicit Cancel Location request should be sent to GMLC in the following scenarios: <ul style="list-style-type: none"> when the <code>maximumNumberOfReports</code> are received from GMLC when Delete Subscription request is received from AF based on the GMLC initiated Cancel Location request.
monitoringevents.gmlc.gmlchAccuracy	Specifies the horizontal accuracy to select eLCS
monitoringevents.gmlc.gmlcVaccuracy	Specifies the vertical accuracy to chose eLCS
fivegcagent.gmlc.baseUrl	The base URL of GMLC
fivegcagent.gmlc.externalClientType	Default value to be sent to GMLC in externalClientType parameter in ProvideLocation Request.
fivegcagent.gmlc.reportingInterval	Indicates the time interval between each event report in seconds.

Observe

NEF provides the metrics for observing GMLC Based Location Monitoring feature. For more information about the metrics, see [NEF Metrics](#).

Maintain

The NEF logs include the GMLC Based Location Monitoring information for the requests received or responses sent by NEF. To get this information, the log levels must be set to **debug**.

The logs messages for the feature contain the `GMLCRequestProcessing` prefix.

The following table lists the logs that can be used for verifying the GMLC Based Location Monitoring functionality:

Table 3-9 Logs to Verify GMLC Based Location Monitoring Functionality

Description	Log Message	Log Level
NEF Identifies that the request is for GMLC	GMLCRequestProcessing: LocationProvide identified as GMLC	INFO
Request translated from T8 to GMLC format	GMLCRequestProcessing: Request translated to GMLC (Translated request)	INFO
Response received from GMLC	GMLCRequestProcessing: Response received (Response)	INFO
GMLC not reachable	GMLCRequestProcessing: Communication with GMLC failed, (Error details)	ERROR
Request to be handled by UDM	GMLCRequestProcessing: Forwarding the subscription request to UDM on GMLC failure	INFO

Table 3-9 (Cont.) Logs to Verify GMLC Based Location Monitoring Functionality

Description	Log Message	Log Level
Cancel Initiated by GMLC	GMLCRequestProcessing: Cancel Location from GMLC	INFO
Cancel Notify sent to AF	GMLCRequestProcessing: Request to cancel subscription sent to AF	INFO

For more information about the logs, see *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

Enable GMLC to UDM Failover

NEF allows you to enable failover to UDM when GMLC subscription fails for a specific reason.

To enable the failover, perform the following configuration using the custom-values.yaml file:

1. Set value of the `gmlc.switchToUdmOnFailure` parameter. For further information on configuring, refer to Update GMLC Configuration section in *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*. The applicable values are:
 - **IMMEDIATE**: To enable the failover for subscription of Immediate Location Request.
 - **DEFERRED**: To enable the failover for subscription of Deferred Location Request.
 - **ALL**: To enable the failover for subscription of both Immediate and Deferred request.
 - **NONE**: To disable the failover.
2. Configure the `gmlc.switchOnErrorCodes` parameter to define the error codes and causes for which the failover must be performed. For further information on configuring, refer to Update GMLC Configuration section in *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.
3. Configure the `global.configurableErrorCodes` parameters for Egress gateway with the same error code details that are provided in the `gmlc.switchOnErrorCodes`. The following code snippet shows an example to set the parameters in the global configuration section to do the failover:

```
global:
  configurableErrorCodes:
    enabled: true
    errorScenarios:
      - exceptionType: "ConnectException"
        errorCode: "503"
        errorDescription: "Connection failure"
        errorCause: "Connection Refused"
        errorTitle: "ConnectException"
      - exceptionType: "UnknownHostException"
        errorCode: "503"
        errorDescription: "Connection failure"
        errorCause: "Unknown Host Exception"
        errorTitle: "UnknownHostException"
```


Note

The values provided for the `gmlc.switchOnErrorCodes` parameters must also be configured in the `global.configurableErrorCodes` configurations of Egress GW to enable the successful failover.

For more information about the configurations, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*

3.9 CAPIF Event Management

The CAPIF Event Management feature enables NEF to notify AFs or other services about the events occurring at CAPIF through the NEF Event Manager service.

As per the 3GPP specifications, the NEF Event Manager service performs the following functionality:

- Supports AFs and other NEF services to subscribe to CAPIF event notifications.
- Sends notifications to the subscribers in case of an event.
- Unsubscribes AFs or NEF services from the CAPIF event notifications.

For more information about the Event Management functionality, see "Subscription, unsubscription and notifications for the CAPIF events" in 3GPP Technical Specification 23.222, Release 16.

Supported Events

The CAPIF Event Manager service supports subscription for following CAPIF events:

Table 3-10 Supported CAPIF Events

Event Name	Description
SERVICE_API_AVAILABLE	Events related to the availability of a service API after the service APIs are published. Consumer: AF Producer: API Manager
SERVICE_API_UNAVAILABLE	Events related to the unavailability of a service API after the service APIs are unpublished. Consumer: AF Producer: API Manager
API_INVOKER_ONBOARDED	Events related to API invoker status when the invoker is onboarded to CAPIF Consumer: AEF Producer: AF
API_INVOKER_OFFBOARDED	Events related to API invoker status when the invoker is offboarded from CAPIF Consumer: API Exposing Function (AEF) Producer: AF
SERVICE_API_INVOCATION_SUCCESS	Events related to the successful invocation of service APIs Consumer: Audit Service Producer: Security Manager

Table 3-10 (Cont.) Supported CAPIF Events

Event Name	Description
SERVICE_API_INVOCATION_FAILURE	Events related to the failed invocation of service APIs Consumer: Audit Service Producer: Security Manager
API_INVOKER_AUTHORIZATION_REVOKED	Events related to the revocation of the authorization of API invokers to access a service API Consumer: Security Manager Producer: OAM

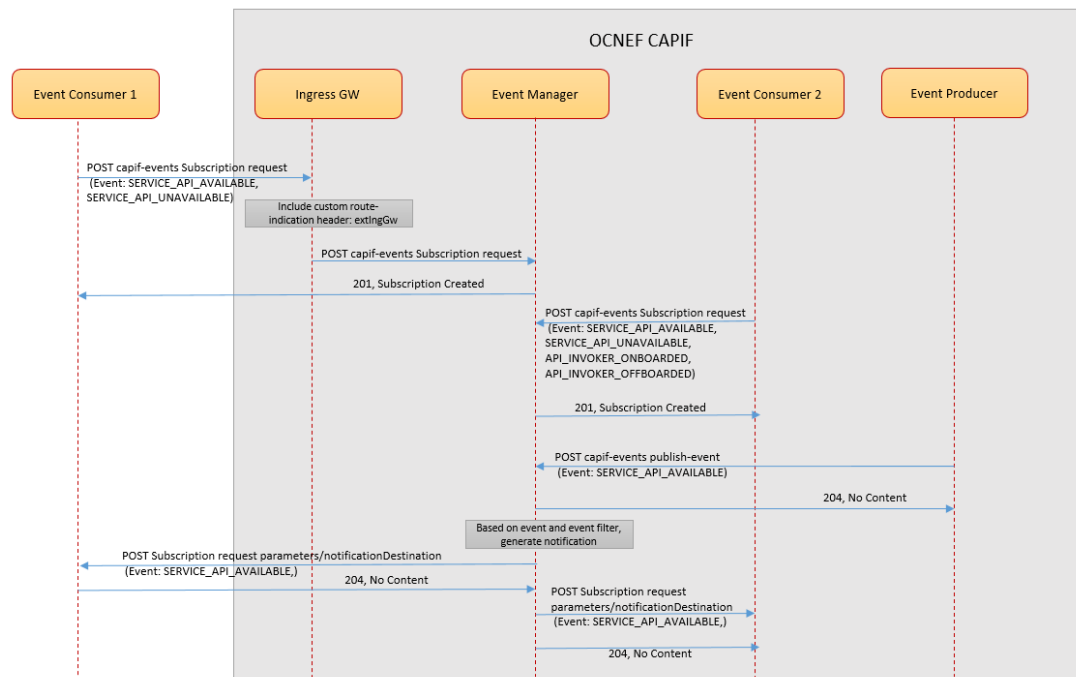
Note

Currently, only for the `API_INVOKER_OFFBOARDED` event, subscription, unsubscription, and notification is supported. For other events, only subscription and unsubscription is supported.

Example- Subscribing to Event

The following diagram shows a high-level call flow where the Event Manager service manages the subscription requests coming from the different event's consumers and sends notifications based on the event occurred at the producer:

Figure 3-19 Call Flow for CAPIF Event Management- Subscribe



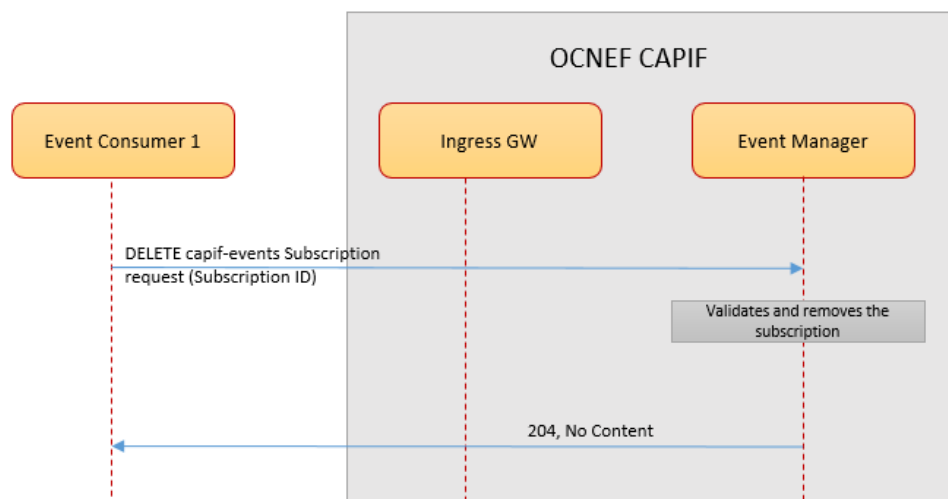
The call flow can be described as follows:

1. An external event consumer (Event Consumer1) sends a POST `capif-events` subscription request to the Event Manager service through the external Ingress GW in NEF deployment.
2. The Event Manager service authenticates the request and creates a subscription.
3. Similarly, an internal NEF service (Event Consumer2) sends a POST `capif-events` subscription request to Event Manager and creates an event notification subscription.
4. The Event Manager receives a POST `capif-events` publish request from the Event Producer.
5. The Event Manager processes the request and based on the parameter values received in the subscription requests, sends the notifications to both Event Consumer1 and the Event Consumer2 for the requested event.

Example- Unsubscribing Event

The following diagram shows a high-level call flow where the Event Manager service manages the delete subscription request coming from an event consumer and deletes the subscription:

Figure 3-20 Call Flow for CAPIF Event Management - Unsubscribe



The call flow can be described as follows:

1. A subscriber (Event Consumer1) sends a DELETE `capif-events` subscription request to the Event Manager service through the external Ingress GW in NEF deployment.
2. The Event Manager service processes the request and validates the subscriber identity.
3. After successful validation, the Event Manager deletes the subscription from NEF database.

For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Managing CAPIF Event Management

Enable

The CAPIF Event Management is a core functionality of NEF. It remains enabled by default.

Observe

NEF provides the metrics for observing the CAPIF Event Management feature. For more information about the metrics, see the "Event Manager Metrics" in [CAPIF Metrics](#).

3.10 Support for AF Session with QoS

In network deployments, operators have the requirement to offer services of a certain quality. There are scenarios when operators need to provide different Quality of Services (QoS) to different types of subscribers or UEs.

NEF enables the operators to manage the QoS using a set of parameters related to the traffic performance on networks. It also provides the capability to set up different QoS standards for different UE sessions based on the service requirements and other specifications. To perform this functionality, the NEF QoS service communicates with Policy Control Function (PCF) to set up, modify, and revoke an AF session with the required QoS.

The AF session with the QoS service feature allows AF to request a data session for a UE with a specific QoS.

The AF sends a request to NEF to provide QoS for the AF session using a QoS reference parameter, which refers to the predefined QoS information. NEF authorizes the request and communicates with the PCF. When the PCF authorizes the service information from the AF and generates a PCC rule, it derives the QoS parameters of the PCC rule based on the service information and the indicated QoS reference parameter. The AF may change the QoS by providing a different QoS reference parameter for an ongoing session. If this happens, the PCF updates the related QoS parameter sets in the PCC rule accordingly.

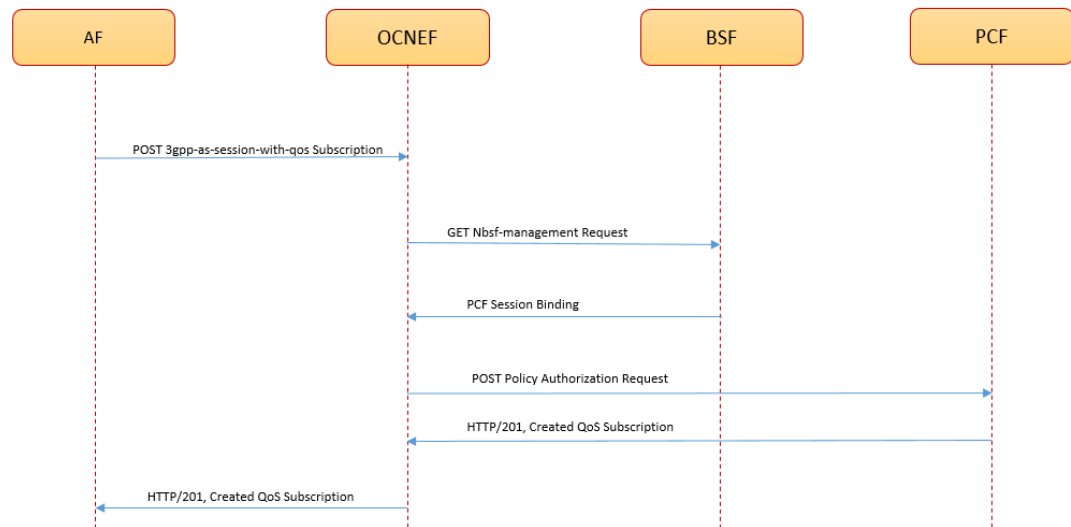
The AF Session with QoS Service functionality enables NEF to perform the following functionality:

- Set up an AF session with the required QoS
- Get the QoS session details for an AF
- Update the QoS subscription for an AF
- Delete an AF session with QoS
- Receive the QoS notifications, such as QoS Monitoring or Usage reports from the PCF and forward it to the subscribed AF

To set up AF sessions with QoS, NEF invokes the PCF that controls the privacy checking of the target subscriber. The invoked PCF authorizes the subscription or notification request, performs the required operation, and sends response to NEF. NEF exposes the information as and when received from the PCF to the AF.

Create Subscription Call Flow

The following diagram is a high-level call flow where NEF receives a subscription request for QoS session from AF and interacts with a PCF to get the relevant information:

Figure 3-21 Call Flow for AF Session with QoS Subscription

The call flow can be described as follows:

1. To set up a session with required QoS, AF sends a 3gpp-as-session-with-qos POST request to NEF.
2. The request routes through the external Ingress Gateway to NEF QoS service after a successful authentication process at the CAPIF.
3. The QoS service performs the following tasks:
 - a. Retrieves the PCF session binding information for the requested UE. The QoS service sends the nbsf-management GET request to the BSF and receives the SUPI, GPSI, PCF Fqdn, and binding level information in response.
 - b. Sends npcf-policyauthorization POST request to the PCF to get the AF request authorized and create policies as requested for the associated PDU session.

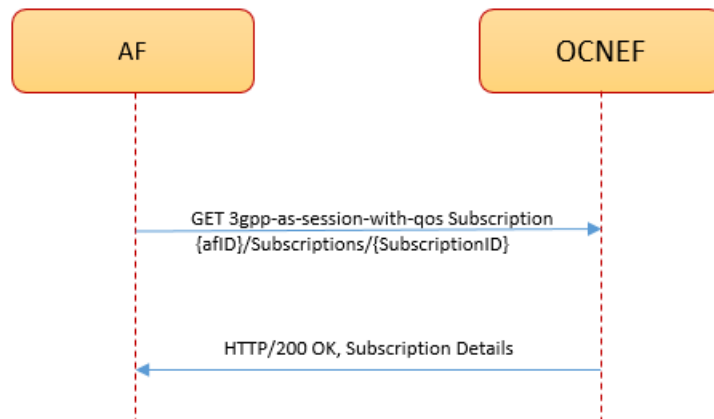
Note

In case, BSF is not enabled then the QoS service sends the npcf-policyauthorization POST request to the PCF as per the pcfBaseUrl parameter value configured in the helm configurations. For more information about the configurations, see [Managing AF Session with QoS](#).

4. On successful processing, PCF creates an individual session resource with the QoS monitoring subscription and sends the HTTP/201 Created response to the 5GC Agent service.
5. NEF forwards the HTTP/201 Created response to the AF.

Get Subscription Call Flow

The following diagram is a high-level call flow where NEF receives a get subscription request for QoS session from AF and processes the request:

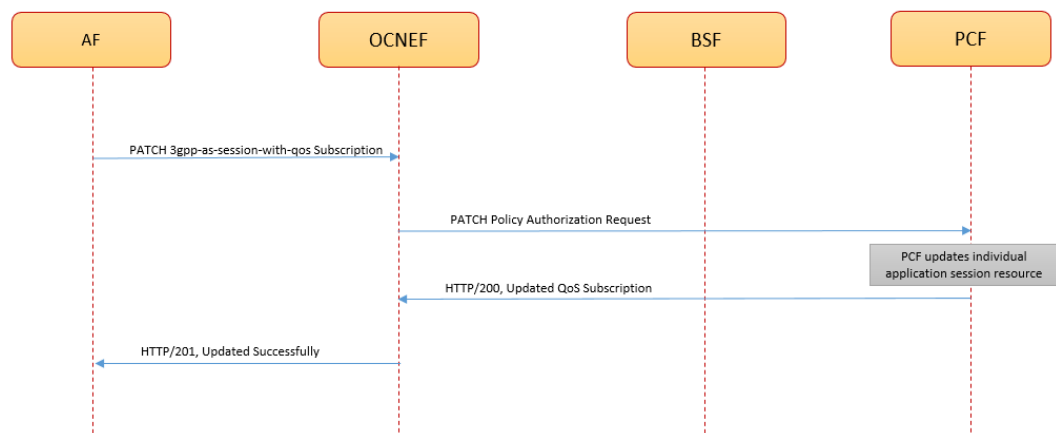
Figure 3-22 Call Flow for GET AF Session with QoS Subscription

The call flow can be described as follows:

1. To retrieve a QoS subscription details AF sends a `3gpp-as-session-with-qos` GET request to NEF.
2. The request routes through the external Ingress Gateway to NEF QoS service after a successful authentication process at the CAPIF.
3. The QoS service retrieves the application session resource from the NEF database and sends the `HTTP/200 OK` response to the AF.

Update Subscription Call Flow

The following diagram is a high-level call flow where NEF receives a PUT or PATCH request for modifying a QoS subscription resource from AF and processes the request:

Figure 3-23 Call Flow for Update AF Session with QoS Subscription

The call flow can be described as follows:

1. To modify a QoS subscription resource, AF sends a `3gpp-as-session-with-qos` PATCH request to NEF.

Note

NEF allows AFs to perform both PATCH and PUT operations to modify the QoS subscription resources.

2. The request routes through the external Ingress Gateway to NEF QoS service after a successful authentication process at the CAPIF.
3. The QoS service sends `npcf-policyauthorization` PATCH request to the PCF to get the AF request authorized and update policies as requested for the associated PDU session.
4. On successful processing, PCF updates the session resources with the QoS monitoring subscription and sends the `HTTP/201 Updated` response to the 5GC Agent service.
5. NEF forwards the `HTTP/201 Updated` response to the AF.

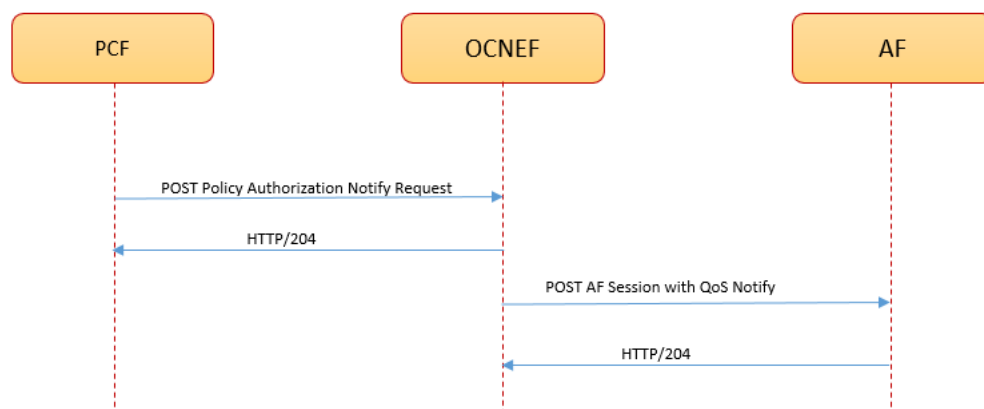
Note

The `npcf-policyauthorization` REST API supports only PATCH operation. As a result, the Update QoS Subscription (PUT or PATCH) operation does not support the removal of any existing parameter. It only supports the addition of new parameters or updating the value of an existing parameter.

Notification Call Flow

The following diagram shows a high-level call flow where NEF receives a QoS notification report from PCF and forwards it to the AF:

Figure 3-24 Call Flow for QoS Event Notification



The call flow can be described as follows:

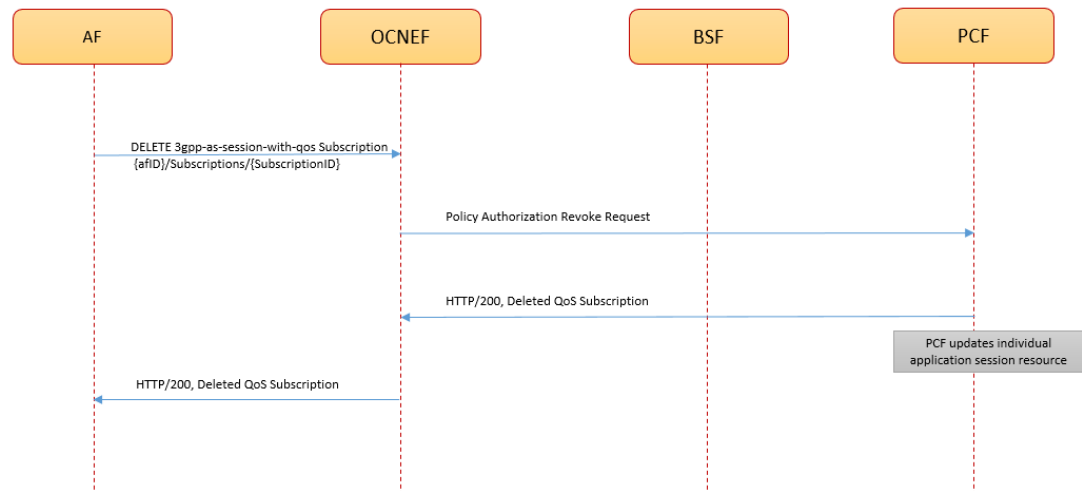
1. PCF sends the QoS notifications `npcf-policyauthorization` notify POST request to NEF with the notification destination information. The request routes through the 5GC Ingress Gateway and 5GC Agent services to QoS Service.
2. The QoS Service processes the request and sends `HTTP/204` response to the PCF.

3. The QoS service forwards the POST request with the notification to the corresponding AF. This sends the QoS notification report of the specified UE to the AF.

Delete Subscription Call Flow

The following diagram is a high-level call flow where NEF receives a request to delete a QoS subscription from AF and interacts with a PCF to process the request:

Figure 3-25 Call Flow for Delete AF Session with QoS Subscription



The call flow can be described as follows:

1. To delete a session with required QoS, AF sends a 3gpp-as-session-with-qos DELETE request to NEF.
2. The request routes through the external Ingress Gateway to NEF QoS service after a successful authentication process at the CAPIF.
3. The QoS service sends the npcf-policyauthorization DELETE request to the PCF.
4. On successful processing, PCF deletes the session resource subscription and sends the HTTP/200 OK or HTTP/204 No Content response to NEF.
5. NEF forwards the response to the AF.

After successful deletion, the AF stops receiving the QoS event notifications.

For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Managing AF Session with QoS

Enable

AF Session with QoS is a core functionality of NEF. It remains enabled by default. You must configure the `qualityofservice` parameters for QoS service in the `ocnef-custom-values.yaml` file.

Configure

This section explains the configuration parameters required for this feature.

For setting up the QoS parameters requested by AF, NEF communicates with PCF and BSF using both indirect and direct communication models.

Note

NEF allows `AFSessionWithQoS` procedure to support the `mediaType` parameter in the PCF `Npcf_PolicyAuthorisation` API. For more information on this API, see *Oracle Communications Cloud Native Core, Policy User Guide*.

The configurations required are as follows:

Set up the Base URL for PCF and BSF

You can configure the base URL values for BSF and PCF by updating the following parameters in the `ocnef-custom-values.yaml`:

- `fivegcagent.pcfBaseUrl`
- `fivegcagent.bsfBaseUrl`

Enable BSF

NEF allows you to obtain address information of the selected PCF for a PDU session through the BSF.

To set up a communication with PCF through BSF, set the **bsfEnabled** parameter to **true**, under the `fivegcagent` service configurations in the `ocnef-custom-values.yaml` file.

Configure Direct Communication with PCF

To set up a direct communication model with PCF, configure the following parameters for setting up the Model A or Model B - Direct Communication models for PCF under the `fivegcagent` configurations:

Table 3-11 `fivegcagent` Configuration

Parameter	Description
<code>pcfBaseUrl</code>	Base URL of the PCF with which NEF communicates.
<code>targetNfCommunicationProfileMapping</code>	The supported communication method for PCF. Possible values are: <ul style="list-style-type: none">• <code>model A</code>• <code>model B</code>

Note

In the case of Model B - Direct communication, add **BSF** and **PCF** to the `nrfClientSubscribeTypes` parameter under the **nrfclient** configurations:

Configure Indirect Communication with PCF

Create the Model D communication profiles for BSF and PCF using the **communicationProfiles** configurations under the `fivegcagent` configurations.

Table 3-12 Configurations for Model D Indirect Communication

Parameter	Description
customBSFModelD.discoveryHeaderParams.targetNfType	The target NF, with which NEF is going to have the indirect communication.
customBSFModelD.discoveryHeaderParams.discoveryServices	The service names for the discovery NF.
customBSFModelD.discoveryHeaderParams.supportedFeatures	This parameter is mapped with the 3gpp-Sbi-Discovery-supported-features discovery header
customBSFModelD.sendDiscoverHeaderInitMsg	Flag to control whether to send discovery headers in initial message or not
customBSFModelD.sendDiscoverHeaderSubsMsg	Flag to control whether to send discovery headers in subsequent message or not
customBSFModelD.sendRoutingBindingHeader	Indicates if the routing binding header must be included or not
customPCFModelD.discoveryHeaderParams.targetNfType	The target NF, with which NEF is going to have the indirect communication.
customPCFModelD.discoveryHeaderParams.discoveryServices	The service names for the discovery NF. This parameter is mapped with the 3gpp-Sbi-Discovery-service-names header
customPCFModelD.discoveryHeaderParams.supportedFeatures	This parameter is mapped with the 3gpp-Sbi-Discovery-supported-features discovery header
customPCFModelD.sendDiscoverHeaderInitMsg	Flag to control whether to send discovery headers in initial message or not
customPCFModelD.sendDiscoverHeaderSubsMsg	Flag to control whether to send discovery headers in subsequent message or not
customPCFModelD.sendRoutingBindingHeader	Indicates if the routing binding header must be included or not
targetNfCommunicationProfileMapping.BSF	The supported communication method for PCF. Value: model D
targetNfCommunicationProfileMapping.PCF	The supported communication method for PCF. Value: model D
customModel.discoveryHeaderParams.preferredLocality	Preferred target NF location for the discovery NF. This parameter is mapped with 3gpp-Sbi-Discovery-preferred-locality.

For more information about the configurations, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Once the parameter values are set, you can configure the session with the NEF that has the required QoS using the REST API for NEF. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide*.

Observe

NEF provides the Quality of Service related metrics for observing AF Session with QoS feature. For more information about the metrics, see [NEF Metrics](#).

3.10.1 Converged SCEF NEF for QoS

The Converged SCEF-NEF for Quality of Services (QoS) feature sets up an AF Session with the required QoS in the 4G system as a fallback solution in case AF sessions with QoS subscription fail with the 5G system. The same AFSessionWithQoS API is used in the 4G system to set up an AS session with the required QoS for the service for 4G.

When NEF receives an Nnef_AFSessionwithQos request from AF, it initiates N5 based QoS call flow with PCF+PCRF. If PCF+PCRF returns an error response, that indicates that the incoming request for a specific PDU session is not for 5G [example, 404 error response], then NEF initiates the Rx based QoS call flow with the same PCF+PCRF a 4G diameter host of PCF.

If in case the UE is latched to a 4G network, then NEF cannot create subscription or receive notification from that network. NEF needs to be enabled with the Converged SCEF-NEF solution for specific services where interaction between the Service Capability Exposure Function (SCEF) and NEF is required.

When there is a UE capable of mobility between Evolved Packet Core (EPS) and 5G Core (5GC), the network is expected to associate the UE with an SCEF+NEF node for Service Capability Exposure.

With the introduction of this feature, when the AFSessionWithQos fails in the PCF authorization flow towards PCF NF, NEF attempts to create subscription towards PCRF if the failure response from PCF matches the configured `qualityofservice.switchOnErrorCodes.code` and `qualityofservice.switchOnErrorCodes.cause` values. The `qualityofservice.switchToPCRFOnAuthFailure` flag must be enabled for this feature.

Note

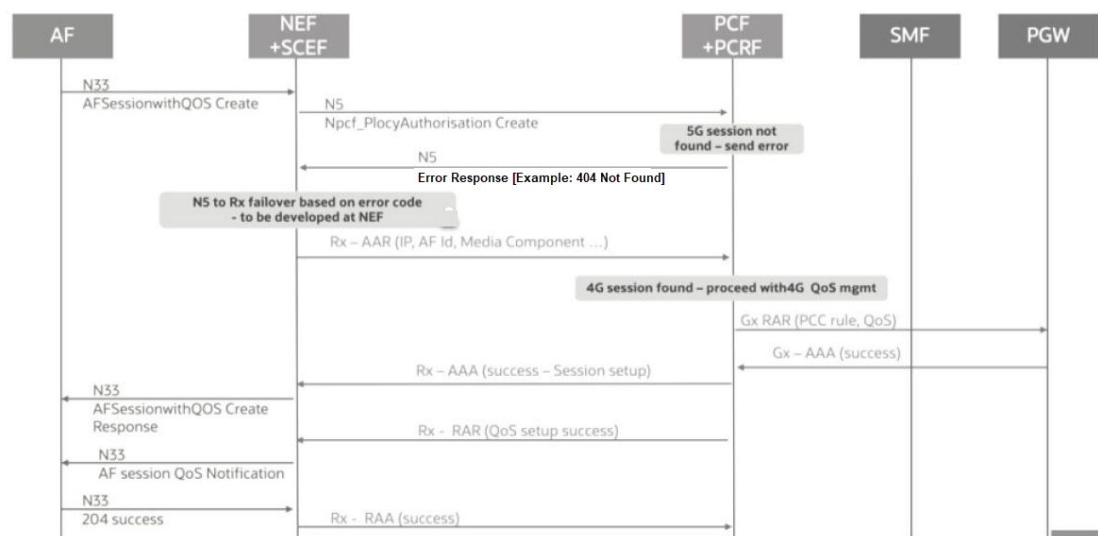
- This switchover based on configured code and cause values is not applicable if the AFSessionWithQos has 5G specific attributes.
- The remaining notification flow is similar to the existing flow from NEF to AF.

Converged SCEF-NEF feature allows NEF to communicate with 4G network nodes using the diameter based southbound interfaces.

AFSessionWithQos Converged SCEF-NEF Call Flow

The following diagram is a high-level call flow where NEF+SCEF sends a create request for QoS session from AF and interacts with a PCF to get the relevant information:

Figure 3-26 Call Flow for Error Code based N5 to Rx Failover for 4G Subscriber



The call flow can be described as follows:

1. To create a session with required QoS, AF sends a AFSessionwithQOS Create request to NEF+SCEF.
2. NEF sends Npcf_PolicyAuthorization_Create request to PCF.
3. PCF rejects the request with an error code. This is caused here due to a PDU session being not found.
4. If NEF is enabled for Converged SCEF NEF QoS and failover configurations matches the error response, then AAR diameter request is sent to the configured PCRF peer node over Rx interface.
5. On successful validation of authorization request, PCRF responds with a RAR diameter success response.
6. NEF receives the diameter response and propagates the response back to AF.
7. Whenever PCRF detects an event, it notifies the same in RAR diameter message to NEF.
8. NEF translates and forwards it as http notification request to the corresponding AF.
9. AF acknowledges the notification request with 204 http response.

Managing Converged SCEF-NEF

Enable

You can enable the Converged SCEF-NEF service by setting `convergedScefNefEnabled` parameter to true. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Note

The `convergedScefNefEnabled` parameter:

- introduces `Converged_scef_nef` diameter gateway for receiving diameter traffic from EPC network nodes.
- enables EPC subscription parameters for subscriptions towards UDM (currently `monitoringEvents` Location reporting subscriptions).

Configure

You can configure this feature using Helm parameters.

For information about configuring the `convergedScefNefEnabled`, `scefDiamHost`, `envDiameterRealm`, `envDiameterIdentity` and `ocnef-diam-gateway` microservice parameters, see *Oracle Communication Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Observe

NEF provides the Converged SCEF-NEF related metrics and alerts for observing AF Session with this feature. For more information about the metrics, see [NEF Metrics](#). For more information about the alerts, see [Alerts](#).

3.11 Support for Georedundancy

The NEF architecture supports Geographical Redundant (Georedundant) deployments to ensure high availability and redundancy. It offers two-sites georedundancy to ensure service availability when one of the NEF sites is down.

When NEF is deployed in a two-site georedundant setup, both the NEF sites work in an Active state with the following specifications:

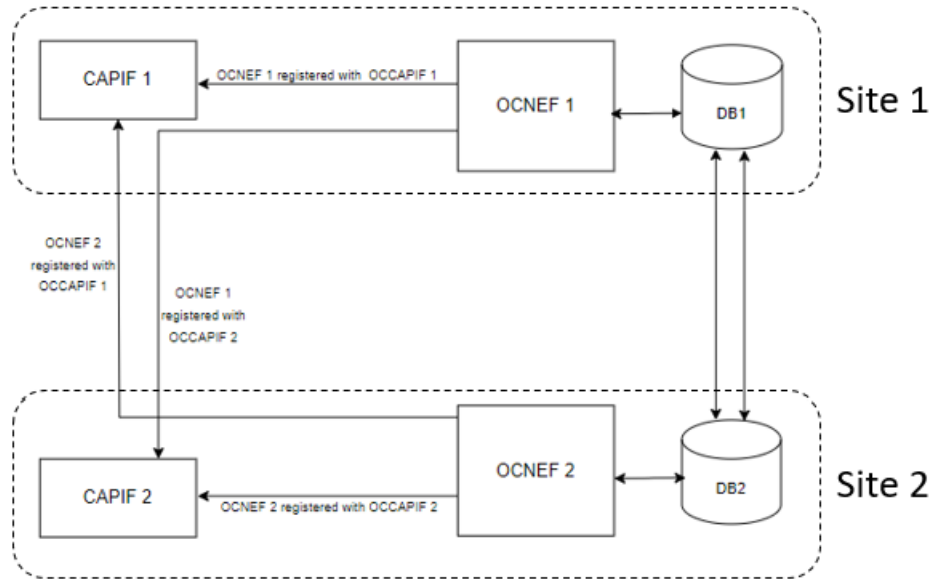
- In case of a site failure, the other NEF site starts serving the NFs or AFs of the failed site. For example in a two-sites georedundant system with Site A and Site B:
 - If Site A fails, NEF (CAPIF and NEF) at Site B starts serving the consumers of Site-A.
 - If Site B fails, NEFs (CAPIF and NEF) at Site A start serving the consumers of Site-B.
- The georedundant NEF deployment supports only Model B and Model D communications. NEF does not support the Model A - direct communication model in a georedundant deployment.
- Both the NEF sites remain active and provide service in the active-active mode.
- The NEF instances share the Session State data using DB tier replication service to enable service continuity during a site failure.
- All the NEF instances register with NRF independently and get notified when another instance is down.

Two-Site Georedundancy Deployment

When NEF is deployed as two-sites georedundant instances of CAPIF and NEF, then:

- Every active site contains one instance of CAPIF and NEF
- Each NEF instance registers with each CAPIF instance

The following diagram depicts the topology for two-site georedundant NEF deployment:

Figure 3-27 Two-Site Georedundancy Deployment

The topology has the following configurations:

Site 1: CAPIF 1 and NEF 1

Site 2: CAPIF 2 and NEF 2

The AFs can onboard the API invokers through any of the CAPIF instances and get access to all the NEF instances. For example, AF onboards the API Invoker on both the instances of CAPIF and receives the OAuth Token1 and OAuth Token2 respectively to access the services of NEF1. This enables the AF to create subscriptions on any of the available NEF instances using either of the access tokens.

Note

The subscription data from the AF must contain the information of the owner site.

All the notifications for the subscriptions are managed at the owner site in case of no site failure. If the notification generating NF, such as UDM or AMF cannot reach the owner site, then the NF retrieves the URL of the georedudant NEF instance and forwards the notification to that site.

Note

In below cases, the GR configuration at NEF checks if peer site is down and then processes notification:

- In case of 5G notification, if the NEF site is down, then the consumer NF queries NRF to get peer NEF and send notification.
- If Converged SCEF-NEF is enabled and the NEF site is down, when notification comes from 4G core, the consumer 4G node may route the request to active NEF site, based on operator configuration or DRA routing policy.

The georedundant NEF site processes the request only if the below conditions are met:

- The consumer NEF ensures that the original subscribing NEF is down. This check is performed only if the value of the `geoRedundancyOptions.handleNotification.checkSiteStatus` parameter is set to **true** in the `ocnef-custom-values.yaml` file.

To keep track of peer NEF status, every NEF instance in the georedundant deployment subscribes with NRF for receiving notifications about the change in the status of peer instances. Based on this subscription, whenever the peer NEF comes up or goes down, NRF notifies all the other instances of the deployment.
- The DB replication must be in good health. This check is performed only if the value of the `geoRedundancyOptions.handleNotification.checkDBReplicationStatus` parameter is set to **true** in the `ocnef-custom-values.yaml` file.

Managing Support for Georedundancy

Deploy

To deploy NEF in a georedundant environment:

1. Set up the replicated `cnDBTier` version 22.3.x or above, on two-sites. For information about installing `cnDBTier`, see "Installing `cnDBTier`" in *Oracle Communications Cloud Native Core, `cnDBTier` Installation Guide*.
2. Deploy NEF over the replicated `cnDBTier` sites. For information about installing and deploying NEF, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Configure

To configure georedundancy:

Configure the georedundancy specific parameters while deploying the CAPIF and NEF instances on the replicated sites.

CAPIF Configurations

Configure the following parameters to enable and configure georedundancy in the custom values file for CAPIF:

Table 3-13 Georedundancy Specific Parameters

Parameter	Description
<code>capifInstanceCld</code>	The unique identifier of the CAPIF instance. This parameter value is unique per site.

NEF Configurations

- **Adding CAPIF instance details in NEF:**
Configure the following parameters to add the CAPIF specific details in the NEF using the custom values file:

Table 3-14 Configure CAPIF Details

Parameter	Description
capifInstanceId	The unique identifier for the CAPIF instance.
type	
host	The hostname for the CAPIF instance.
port	The CAPIF instance port.
scheme	
certificate	
secretName	
secretNamespace	
certificateName	

Along with the CAPIF details of the existing site, the NEF configurations must also include the CAPIF details of the other site. When a new site is added to an existing deployment:

- The NEF configurations of the existing as well as new site must include details of both the CAPIF instances
- Both the NEF instances register and publish their services to both the CAPIF instances
- **Configuring Georedundancy:**
Configure the following parameters to enable and configure georedundancy in the custom values file for NEF:

Table 3-15 Georedundancy Specific Parameters

Parameter	Description
geoRedundancyOptions.featureStatus	Specifies if georedundancy feature must be enabled or not. The value must be set to DISABLED in case of single site. Note: All the parameters under geoRedundancyOptions are applicable only when the value of this parameter is "ENABLED".
geoRedundancyOptions.monitorDBReplicationStatusInterval	Polling interval for monitoring replication status in seconds
geoRedundancyOptions.replicationStatusUri	URI to get replication status
geoRedundancyOptions.maxSecondsBehindRemote	Maximum number of seconds behind the mated site in replication
geoRedundancyOptions.peerGRSiteList	List of the georedundant sites
geoRedundancyOptions.peerGRSiteList.siteName	The name of the site. The name must be same as the cnDBTier site name
geoRedundancyOptions.peerGRSiteList.nefInstanceId	NEF Instance Id in that Mated site

Table 3-15 (Cont.) Georedundancy Specific Parameters

Parameter	Description
geoRedundancyOptions.handleNotification.checkSiteStatus	Specifies if NEF must validate the status of the owner site before processing notification in a non-owner site. If the value is set to false , the notification is processed without validating the status of owner site.
geoRedundancyOptions.handleNotification.checkDBReplicationStatus	Specifies if NEF must validate the replication status before processing notification in a non-owner site. If the value is set to false , the notification is processed without verifying the DB replication status.

3.11.1 Adding a Site to an Existing NEF Deployment

This section describes the procedure to add a site to an existing NEF deployment.

Prerequisites

1. NEF connected to a cnDBTier is up and running. This is referred as Site-1 (CAPIF-1 and NEF-1).
2. The CAPIF, NEF, and cnDBTier versions used for Site-1 installation must be identified and same must be used for adding another site.

Adding a site

1. Install a new cnDBTier. This cnDBTier must act as a georedundant database to the cnDBTier in Site-1. For more information to install a cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier User Guide*.
2. Verify the replication channel between the cnDBTier sites by sending the following request to the dbMonitor service of both the cnDBTier sites. The responses from both the cnDBTier sites must show the status of replication channel as up:

```
curl http://<mysql-db-monitor-service>:8080/db-tier/status/replication/
realtime
```

Sample command:

```
curl http://mysql-cluster-db-monitor-svc:8080/db-tier/status/replication/
realtime
```

Sample output:

```
[{"localSiteName":"Site-1","remoteSiteName":"Site-2","replicationStatus":"U
P","secondsBehindRemote": 0}]
```

3. Install CAPIF on Site-2. For more information on installation procedure, see "CAPIF Installation" in *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.
4. Perform the upgrade for NEF on Site-1 to add the details of CAPIF-2 in the `ocnef-custom-values.yaml` file.

Sample configuration:

```

publicKeyMonitorDelay: 15000
capifDetails:
  - capifInstanceId: f98b05e0-22d8-11ed-861d-0242ac120002
    type: local # this parameter would be used to populate the
    capif_instance_id column for existing records in upgrade to 22.4.0
    host: 10.122.123.123
    port: 8080
    scheme: http
    certificate:
      secretName: certificate-secret
      secretNamespace: *nefK8NameSpace
      certificateName: tmp.cer

  - capifInstanceId: f98b05e0-22d8-11ed-861d-0242ac120003
    type: remote
    host: 10.122.123.223
    port: 8080
    scheme: http
    certificate:
      secretName: certificate-secret
      secretNamespace: *nefK8NameSpace
      certificateName: tmp.cer

```

5. Install NEF on Site-2 by providing the details about both the CAPIF instances in the `ocnef-custom-values.yaml`. For more information on installation procedure, see "NEF Installation" in *Oracle Communications Cloud Native Core, Network Exposure Function Installation and Upgrade Guide*.

3.11.2 Removing a Site to from an Existing Georedundant Deployment

Prerequisites

1. NEF connected to a cnDBTier is up and running. This is referred as Site-1 (CAPIF-1 and NEF-1).

This section describes the procedure to remove a site (Site-2) from an existing georedundant NEF deployment.

1. Offboard all the API invokers from CAPIF of the site that needs to be removed. This deletes all the subscriptions related to the API invokers onboarded on the CAPIF that needs to be removed.
2. Upgrade the NEF instance on the surviving site to remove the details of the CAPIF of the site to be removed.
For more information about upgrade procedures, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation and Upgrade Guide*.
3. Uninstall the CAPIF instance from the site to be removed.

! Important

The CAPIF instance removed from a georedundant deployment must not be used in any other NEF deployments or as an standalone CAPIF. It is recommended to uninstall the CAPIF and then install again.

4. Uninstall the NEF instance from the site to be removed.
For more information on uninstallation procedures, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.
5. Remove the DB Replication and the NDBCluster on the site to be removed.

Example:

The following steps provide an example to remove Site-2 that consists of NEF-2 and CAPIF-2 whereas, Site-1 with NEF-1 and CAPIF-1 is the surviving site:

1. Offboard all the invokers from CAPIF-2 on Site-2.
2. Upgrade NEF-1 to remove details of CAPIF-2 from the `ocnef_customize_values.yaml` file during upgrade on Site-1.
3. Uninstall CAPIF-2 instance from Site-2.
4. Uninstall NEF-2 from Site-2.
5. Remove the DB Replication and the NDBCluster on Site-2.

3.12 Converged SCEF-NEF

Converged SCEF-NEF feature allows NEF to communicate with 4G network nodes using the diameter based southbound interfaces.

i Note

- For information on Converged SCEF NEF for ME service, refer to [Converged SCEF NEF Model for Monitoring Event](#).
- For information on Converged SCEF NEF for QoS, refer to [Converged SCEF NEF for QoS](#).

Managing Converged SCEF-NEF

Enable

You can enable the Converged SCEF-NEF service by setting `enableFeature.convergedScefNef` parameter to true. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

i Note

The `convergedScefNefEnabled` parameter introduces `enableFeature.convergedScefNef` diameter gateway for receiving diameter traffic from EPC network nodes.

Configure

You can configure this feature using Helm parameters.

For information about configuring the `convergedScefNefEnabled`, `scefDiamHost`, `envDiameterRealm`, `envDiameterIdentity` and `ocnef-diam-gateway` microservice parameters, see *Oracle Communication Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Observe

NEF provides the Converged SCEF-NEF related metrics and alerts for observing AF Session with this feature. For more information about the metrics, see [NEF Metrics](#). For more information about the alerts, see [Alerts](#).

3.13 Support for Application Function Influence on Traffic Routing

An Application Function (AF) can send requests to influence Session Management Function (SMF) routing decisions for user plane traffic of a Protocol Data Unit (PDU) session. If the operator does not allow an AF to access the network directly, then that AF uses the Network Exposure Function (NEF) to interact with the 5G Core (5GC). In this case, the AF sends request to the NEF through the Traffic Influence APIs. To support Traffic Influence to receive and process such requests, NEF functionality is enhanced with the Traffic Influence feature.

With this feature, AFs can influence SMF routing decisions on the user plane traffic of a PDU session for the targeted UE or a group of UEs. It allows an external AF to decide the routing profile and the route for data plane from UE to network in a particular PDU session.

The AF requests can influence User Plane Function (UPF) selections and allow routing of user traffic to a local access (identified by a DNAI) to a Data Network. AF can also provide this in the subscription request sent to SMF events. This helps to enrich the end user experience and monitor the network improvement.

The Traffic Influence decides which NF the AF request should interact with, based on the attributes in the received request.

Managing Application Function Influence on Traffic Routing

Enable

You can enable the Traffic Influence service by setting `trafficInfluenceEnabled` parameter to true. For more information, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure this feature using Helm parameters. You can create the Model A, B, and D communication profiles using the `custom-values.yaml` file.

For information about configuring the `targetNfCommunicationProfileMapping`, `fivegcagent`, `communicationProfiles`, and `trafficinfluence` microservice parameters. See *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Observe

NEF provides the Traffic Influence related metrics for observing AF Session with this feature. For more information about the metrics, see [NEF Metrics](#).

3.14 Automated Test Suite Support

NEF provides Automated Test Suite (ATS) for validating the functionalities. ATS allows you to run NEF test cases using an automated testing tool, and then compares the actual results with the expected or predicted results. In this process, there is no intervention from the user. For more information about installing and configuring ATS, see *Oracle Communications Cloud Native Core, Automated Test Suite (ATS) Guide*.

3.15 Support for Kubernetes Resource

3.15.1 Network Policies

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

Without Network Policy in place, the pods under NEF or CAPIF deployment can be contacted by any other pods in the Kubernetes cluster without any restrictions. This could lead to potential security threat. Network Policy provides namespace-level isolation, which allows secured communications to and from NEF or CAPIF with rules defined in respective Network Policies. For example, NEF or CAPIF internal microservices cannot be contacted directly by any other non-NEF or non-CAPIF pods.

Default Network Policies

The following default network policies are provided in NEF and CAPIF network policy Helm charts. If these are changed in NEF or CAPIF deployment, the default ports here must also be updated.

NEF Security Policies:

- **deny-ingress-all-nef:** Blocks all ingress traffic of pods present in a NEF deployment. These pods can be identified using the following labels:

```
"app.kubernetes.io/part-of": ocnef "  
"app.kubernetes.io/part-of": nrf-client "  
"app.kubernetes.io/part-of": ocats-nef "  
"app.kubernetes.io/part-of": ocnef-simulator "
```
- **allow-ingress-sbi-nef:** Allows traffic on the Internet Gateway (IGW) pods on ports 8081 and 8443, which permits SBI traffic
- **deny-egress-all-nef:** Blocks all egress traffic of pods having the following labels:

```
"app.kubernetes.io/part-of": ocnef "  
"app.kubernetes.io/part-of": nrf-client "  
"app.kubernetes.io/part-of": ocats-nef "  
"app.kubernetes.io/part-of": ocnef-simulator "
```
- **allow-egress-egw-nef:** Allows all egress for egress gateways.
- **allow-ingress-prometheus-nef:** Allows the traffic flow from Prometheus service to the NEF with default ports.

- **allow-egress-db-nef:** Allows the traffic to flow from NEF to db sql port and db monitoring port with default ports.
- **allow-egress-k8s-api-nef:** Allows the traffic flow from NEF to Kubernetes API server port.
- **allow-egress-jaeger-nef:** Allows the traffic flow from NEF to Jaeger agent port on default ports.
- **allow-egress-dns-nef:** Allows the traffic flow from NEF to k8s DNS service with default ports.
- **allow-ingress-from-pods-nef:** Allows ingress communication between the different microservices of the NEF.
- **allow-egress-to-pods-nef:** Allows egress communication between the different microservices of the NEF.
- **allow-nodeport-for-nef-ocats:** Allows accessing the NEF ATS pods from the GUI on default port 8080. By default, this policy is commented, uncomment to access the deployed ATS GUI.

CAPIF Security policies:

- **deny-ingress-all-capif:** Blocks all ingress traffic of pods present in a CAPIF deployment. Pods can be identified using the following label:

app.kubernetes.io/part-of: occapif

- **allow-ingress-sbi-capif:** Allows traffic on the IGW pods on ports 8081 and 8443, which permits SBI traffic.
- **deny-egress-all-capif:** Blocks all egress traffic of pods having the following label:

"app.kubernetes.io/part-of: occapif"

- **allow-egress-egw-nef:** Allows all egress for egress gateways.
- **allow-ingress-prometheus-capif:** Allows the traffic flow from Prometheus service to the CAPIF with default ports.
- **allow-egress-db-capif:** Allows the traffic flow from CAPIF to db sql port and db monitoring port with default ports.
- **allow-egress-k8s-api-capif:** Allows the traffic flow from CAPIF to Kubernetes API server port.
- **allow-egress-jaeger-capif:** Allows the traffic flow from CAPIF to Jaeger agent port on default ports.
- **allow-egress-dns:** Allows the traffic flow from CAPIF to k8s DNS service with default ports.
- **allow-ingress-from-pods-capif:** Allows ingress communication between the different microservices of the CAPIF.
- **allow-egress-to-pods-capif:** Allows egress communication between the different microservices of the CAPIF.

Managing Support for Network Policies

Enable

To use this feature, network policies need to be applied to the namespace wherein NEF or CAPIF is applied.

Configure

You can configure this feature using Helm. For information about Configuring Network Policy for NEF Deployment, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no specific metrics and alerts required for the Support of Network Policy functionality.

4

Configuring Network Exposure Function using the CNC Console

This chapter provides information about how to configure and modify different services in Network Exposure Function (NEF) using the Oracle Communications Cloud Native Configuration Console (CNC Console).

The REST API configurations can also be performed using the CNC Console.

Note

While installing CNC Console, configure helm config section as the following:

- `instances` field to be configured with NEF and CAPIF type.
- `fqdn` field to be configured with `fivegc-ingress-gateway` for NEF and `network-ingressgateway` for CAPIF.
- `port` field to be configured with corresponding http1.0 port info
- `owner` field of other configs to be configured with `id`

4.1 Support for Multicuster Deployment

The CNC Console supports both single and multiple cluster deployments.

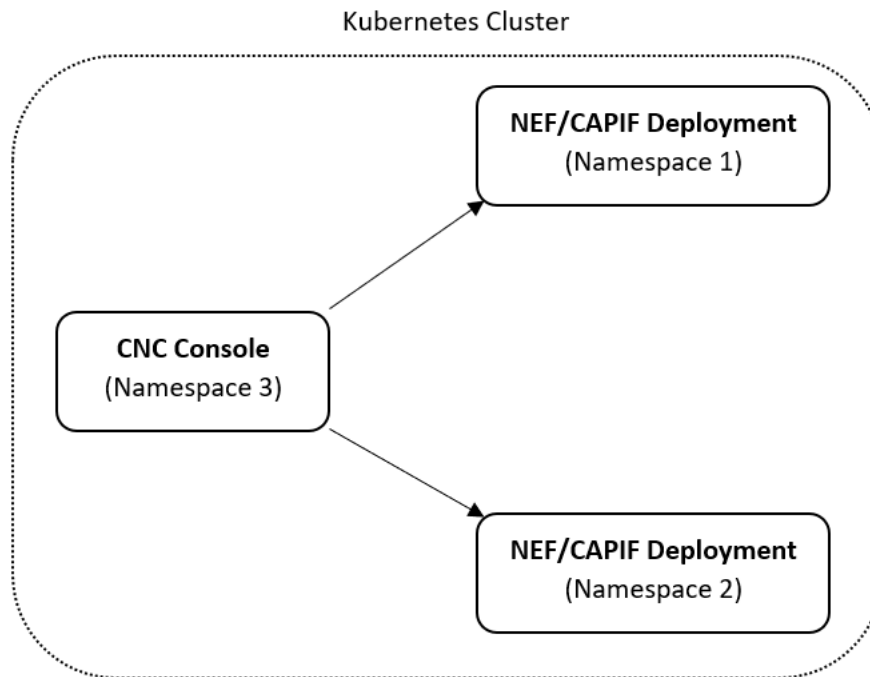
In a single cluster deployment, the CNC Console can manage NFs and *Oracle Communications Cloud Native Environment (OCCNE)* common services deployed in the local Kubernetes clusters.

In a multicuster deployment, the CNC Console manages NFs and OCCNE common services deployed in the remote Kubernetes clusters. For more information about single and multiple cluster deployments, see *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

With multicuster deployment, a single instance of the CNC Console can configure two or multiple instances of NEF deployments if both CNC Console and NEF/CAPIF instances are deployed in the same Kubernetes cluster with different namespaces.

The following image represents a Kubernetes cluster with one instance of CNC Console and two instances of NEF/CAPIF. The single instance of the CNC Console is configuring two instances of NEF/CAPIF with different namespaces.

Figure 4-1 Support for Multicluster Deployment



4.2 CNC Console Interface

This section provides an overview of the CNC Console to configure NEF and CAPIF features.

You can use NEF integrated with the CNC Console after logging in to CNC Console. To log in to the CNC Console, you must make the following updates to the hosts file at the C:\Windows\System32\drivers\etc location.

1. In the Windows system, open the hosts file in the notepad as an Administrator and append the following set of lines at the end:

```
<CNCC Node IP> cncc-iam-ingress-gateway.cncc.svc.cluster.local
<CNCC Node IP> cncc-core-ingress-gateway.cncc.svc.cluster.local
```

Example:

```
10.75.212.88 cncc-iam-ingress-gateway.cncc.svc.cluster.local
10.75.212.88 cncc-core-ingress-gateway.cncc.svc.cluster.local
```

Note

The IP Address in the above lines may change when deployment cluster changes.

2. Save and close the hosts file.
Before logging in to the CNC Console, create a CNC user and password. Log in to the CNC Console using the same credentials. For information about creating a CNC Console

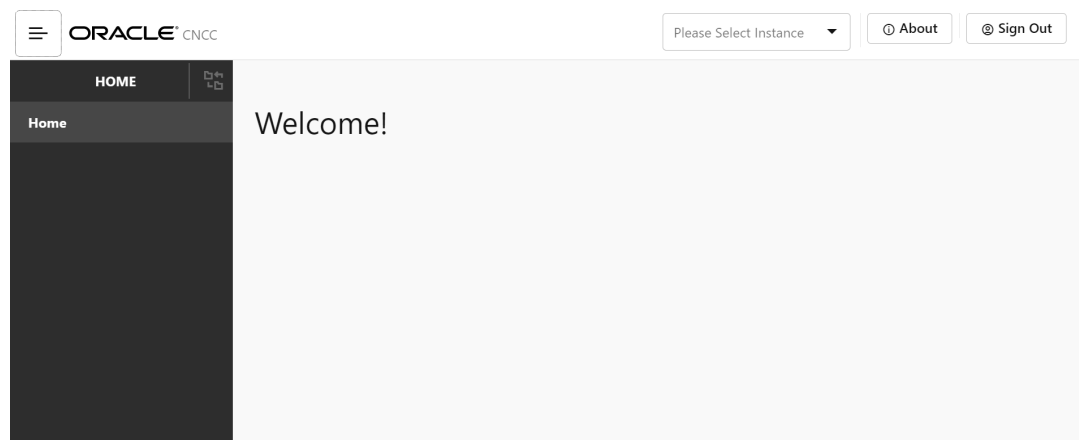
user and password, see *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

Logging in to the CNC Console and Selecting an NEF Instance

Perform the following procedure to log in to the CNC Console and select the required NEF instance to configure NEF features.

1. Open any web browser.
2. Enter the URL: `http://<host name>:<port number>`.
Where, <host name> is `cncc-iam-ingress-ip` and <port number> is `cncc-iam-ingressport`.
3. Enter the login credentials.
4. Click **Log in**.
The CNC Console Home page appears.

Figure 4-2 CNC Console Welcome Screen



5. In the upper pane, from the **Please Select Instance** drop-down list, select the required NEF instance.
The NEF tab appears in the left navigation pane.

Figure 4-3 Select NF Instance

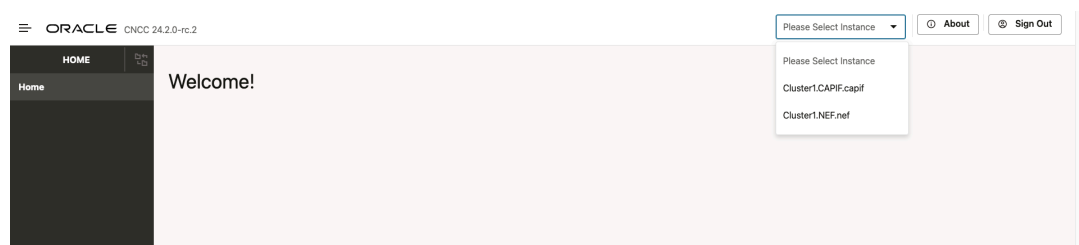
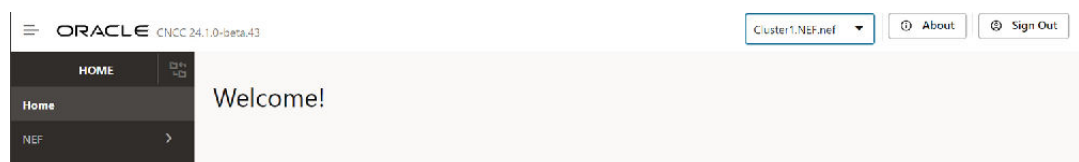


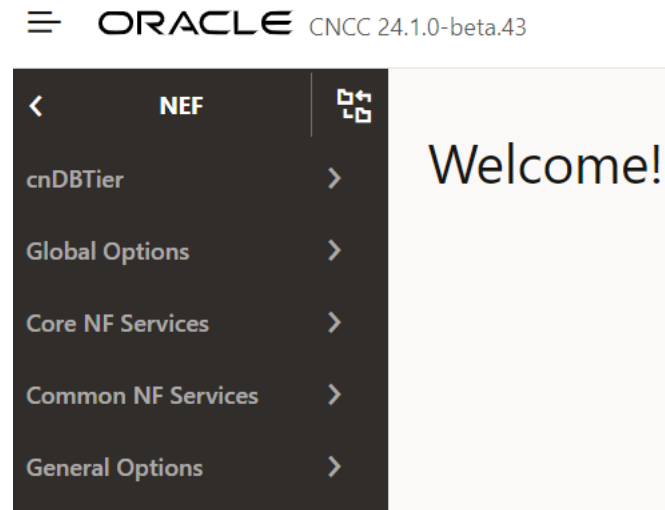
Figure 4-4 NF Instance for NEF



The **Please Select Instance** drop-down list provides NF instances to configure corresponding NF features. You must select an appropriate NEF instance to configure NEF features. Alternatively, you can click one of the following interface elements on the Welcome screen:

- **About:** This element provides the CNC Console product name and version.
- **Sign Out:** This element exits the CNC Console.

Figure 4-5 NF Instances



4.2.1 Configuring NEF Features

This section provides information about enabling the following features of NEF:

Note

You must log in to the CNC Console while performing the procedures described in the subsequent subsections.

- Configuring log level for various services
- Configuring AF Service ID mapping
- Configuring QoS Reference Profile
- Configuring AF ID mapping
- Configuring GMLC options
- Configuring QoS options
- Configuring Short or Long Code for MSISDNless MO SMS
- Configuring SCS Short Messaging Entity
- Configuring PLMN Id mapping
- Configuring TAI mapping
- Configuring ECGI mapping

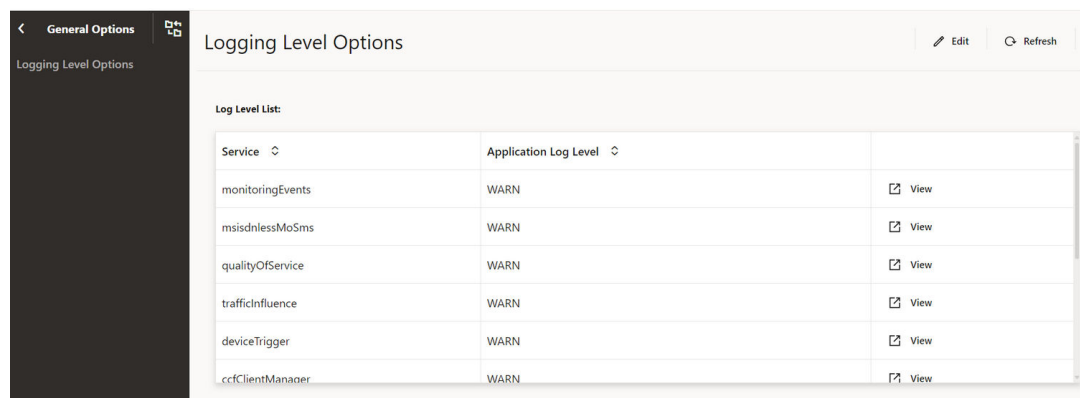
- Configuring NCGI mapping
- Configuring GNB ID mapping
- Configuring Global RAN Node ID mapping
- Configuring Geo Zone Id To Spatial Validity mapping

4.2.1.1 Configuring Log Level for Services

Perform the following procedure to update log level for various services.

1. On the left navigation pane, click the **NEF** tab, then **General Options**, and then **Logging Level Options** tab.
The current list of services in NEF deployment and the corresponding log levels appears.

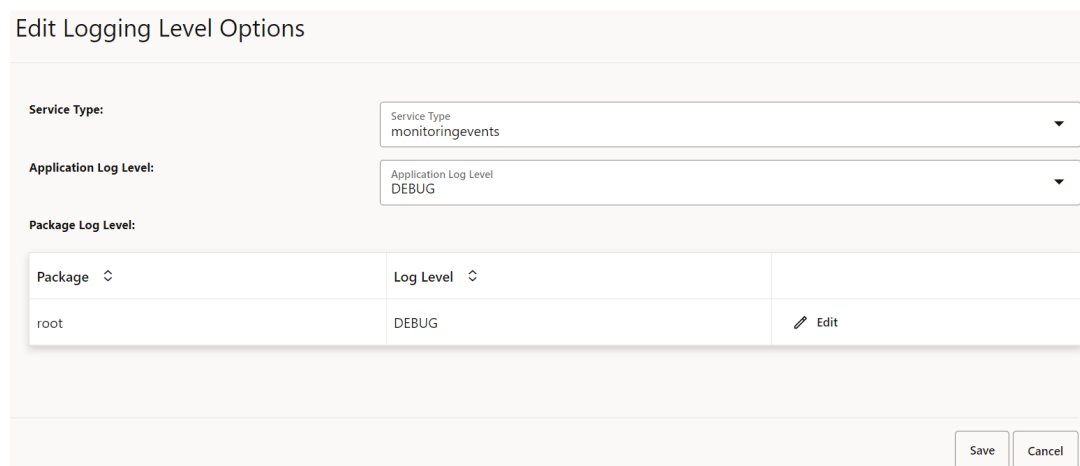
Figure 4-6 Logging Level Options



Service	Application Log Level	
monitoringEvents	WARN	View
msisdnlssMoSms	WARN	View
qualityOfService	WARN	View
trafficInfluence	WARN	View
deviceTrigger	WARN	View
ccfClientManager	WARN	View

2. Click **Edit** to change log level of a specific service type.
The **Edit Logging Level Options** page appears.

Figure 4-7 Edit Logging Level Options



Service Type: monitoringevents

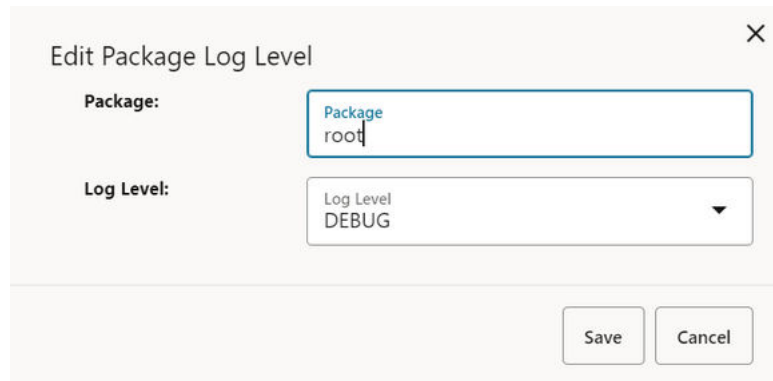
Application Log Level: DEBUG

Package Log Level:

Package	Log Level	
root	DEBUG	Edit

[Save](#) [Cancel](#)

3. Select the required **Service Type** and the corresponding **Application Log Level**.
4. Click **Edit** to change the **Package** and **Log Level** of the corresponding Service Type.

Figure 4-8 Edit Package Log Level

Dialog box titled "Edit Package Log Level" with a close button (X) in the top right corner.

Fields:

- Package:** Text input field containing "Package root".
- Log Level:** Dropdown menu showing "Log Level DEBUG".

Buttons: "Save" and "Cancel" at the bottom right.

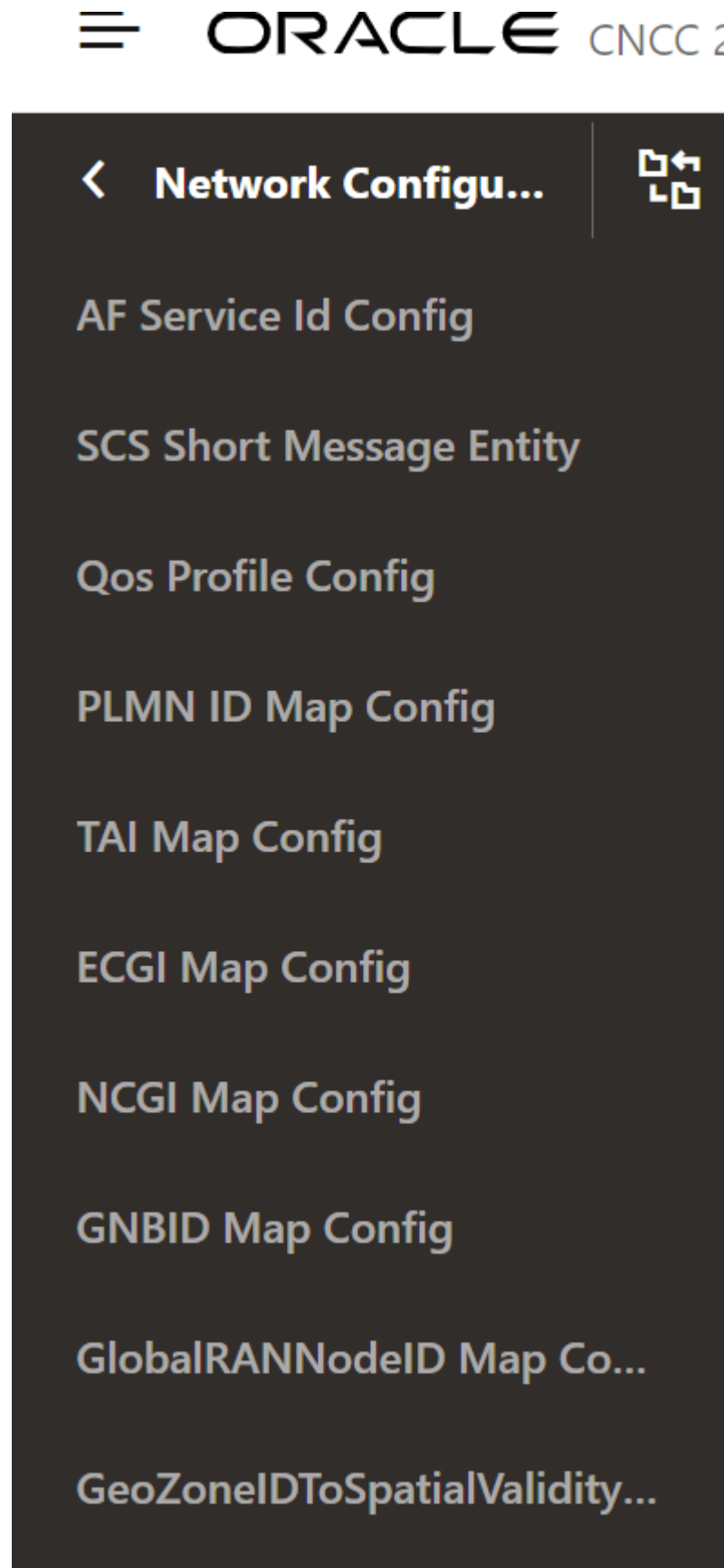
5. Click **Save**.

4.2.1.2 Configuring AF Service ID Mapping

Perform the following procedure to add AF service Id configuration applicable for Traffic Influence feature.

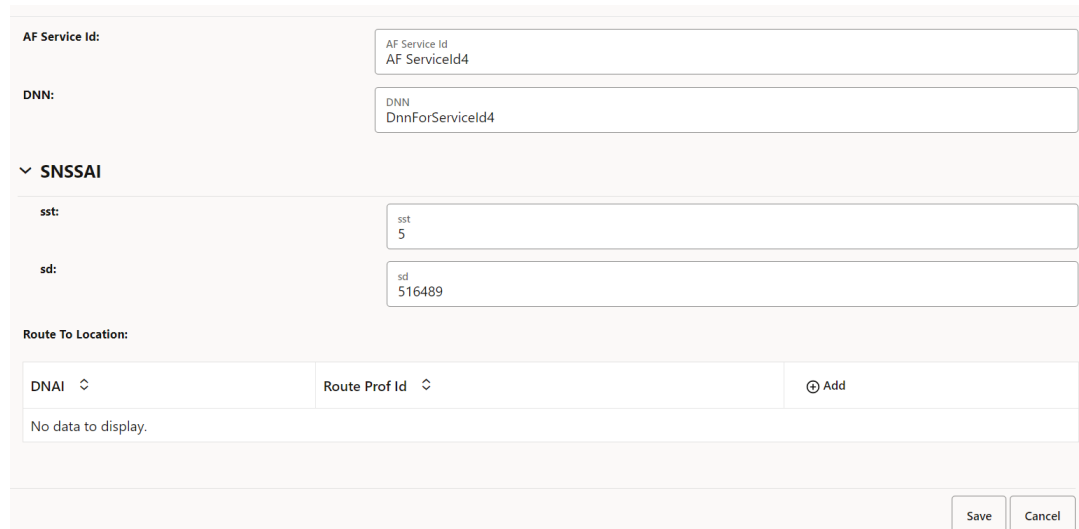
1. On the left navigation pane, click the **NEF** tab, then **Global Options**, and then **Network Configuration** tab.

Figure 4-9 Network Configuration



2. Click **AF Service Id Config**.
3. Click **Add** to create new config entry for a AF Service ID.
4. Update AF Service Id along with the corresponding DNN and Single Network Slice Selection Assistance Information (S-NSSAI) field values appropriately.

Figure 4-10 AF Service Id Edit Page

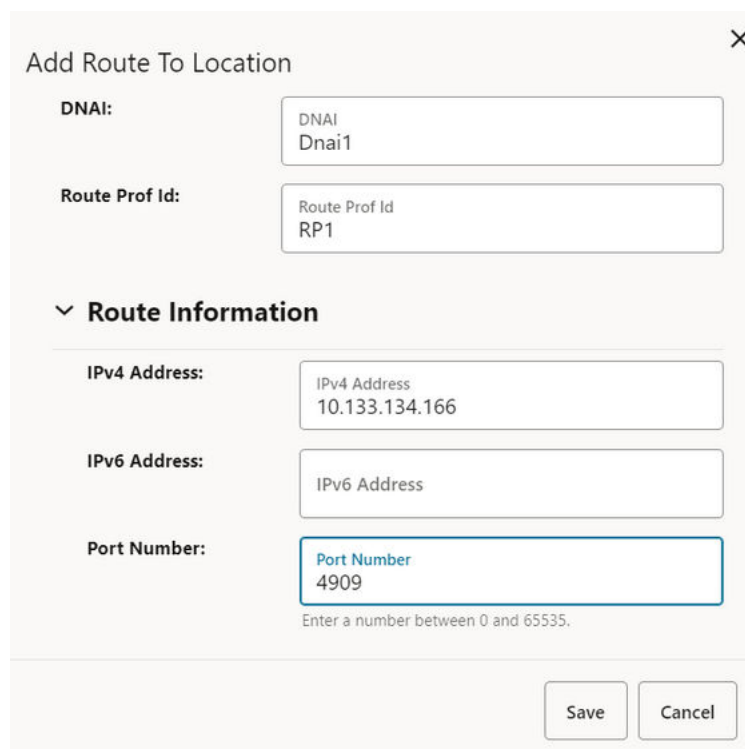


The screenshot shows the 'AF Service Id Edit Page' in the CNC Console. It features several input fields for configuration:

- AF Service Id:** A text field containing 'AF Service Id' and 'AF Serviceld4'.
- DNN:** A text field containing 'DNN' and 'DnnForServiceld4'.
- S-NSSAI:** A section with a dropdown arrow, containing two text fields:
 - sst:** A text field containing 'sst' and '5'.
 - sd:** A text field containing 'sd' and '516489'.
- Route To Location:** A section with two dropdown menus labeled 'DNAI' and 'Route Prof Id', and an 'Add' button. Below these is a message 'No data to display.'
- Buttons:** 'Save' and 'Cancel' buttons at the bottom right.

5. Click **Save**, then select **Add Route To Location**.
6. In **Add Route To Location** page, add or update Route To Location for DNAI, Route Profile Id along with Route Information.









Figure 4-11 Add Route To Location



The screenshot shows the 'Add Route To Location' dialog box. It contains the following fields and sections:

- DNAI:** A text field containing 'DNAI' and 'Dnai1'.
- Route Prof Id:** A text field containing 'Route Prof Id' and 'RP1'.
- Route Information:** A section with a dropdown arrow, containing three text fields:
 - IPv4 Address:** A text field containing 'IPv4 Address' and '10.133.134.166'.
 - IPv6 Address:** A text field containing 'IPv6 Address'.
 - Port Number:** A text field containing 'Port Number' and '4909'. Below this field is a hint: 'Enter a number between 0 and 65535.'
- Buttons:** 'Save' and 'Cancel' buttons at the bottom right.

Figure 4-12 AF Service Id Configuration

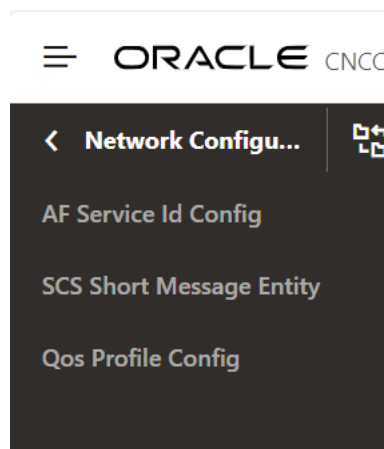
AF Service Id Configuration		
Type to Filter		Refresh Add
AF Service Id	DNN	Actions
afServiceId1	dnnForServiceId1	 
afServiceId2	dnnForServiceId2	 
afServiceId3	dnnForServiceId3	 
AF ServiceId4	DnnForServiceId4	 

- Click **Save**.

4.2.1.3 Configuring QoS Reference Profile



Perform the following procedure to add AF service Id configuration applicable for AF Session with Qos feature.

- On the left navigation pane, click the **NEF** tab, then **Global Options**, and then **Network Configuration** tab.

Figure 4-13 Network Configuration

- Click **QoS Profile Config** to view QoS Profile Configuration.

Figure 4-14 QoS Profile Configuration

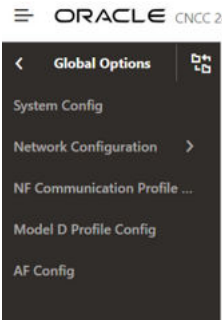
Qos Profile Configuration						
Type to Filter						Refresh
Qos Profile Reference Id	Media Type	Maximum Request BandWidth DL	Maximum Request BandWidth UL	Minimum Request BandWidth DL	Minimum Request BandWidth UL	Actions
qosReferenceId1	AUDIO	100 Gbps	100 Gbps	100 Mbps	100 Mbps	
qosReferenceId2	VIDEO	100 Gbps	100 Gbps	100 Gbps	100 Tbps	

4.2.1.4 Configuring AF ID Mapping

Perform the following procedure to add, update, and delete AF Id mapping configuration.

1. On the left navigation pane, click the **NEF** tab, then **Global Options**.

Figure 4-15 Global Options



2. Click **AF Config** to add AF Id based DNN, SNSSAI config.

Figure 4-16 AF Config Page

AF Config

Type to Filter ✕ Refresh Add

AF Identifier ↕	Actions
No data to display.	

Figure 4-17 AF Identifier Page

AF Identifier:

▼ Config

DNN:

▼ SNSSAI

sst:

sd:

Save Cancel

3. Click **Save**.

Figure 4-18 AF Config Id Added

AF Config	
Type to Filter ✕ ↻ Refresh ⊕ Add	
AF Identifier ↕	Actions
AF Identifier1	✎ 🗑

4.2.1.5 Configuring GMLC Options

Perform the following procedure to add/update/delete GMLC configuration for Monitoring Events feature.

1. On the left navigation pane, click the **NEF** tab, and then **Core NF Services**.

Figure 4-19 Core NF Services

ORACLE CNCC	
< Core NF Services	🗖
Monitoring Events	>
Traffic Influence	>
5GC Agent	>
Quality Of Service	>
AEF API Router	>
Diameter Gateway	>
Device Trigger	>
MSISDNless MO SMS	>

2. Click **Monitoring Events** and then **GMLC options** for ME Service.

Figure 4-20 GMLC Config

GMLC Config

Is GMLC Enabled:

Is GMLC Enabled
false

Note

Edit GMLC operation is only valid when GMLC is enabled.

Incase if GMLC is enabled during installation, then further GMLC config is enabled.

Figure 4-21 GMLC Enabled

GMLC Config

[Edit](#)
[Refresh](#)

Is GMLC Enabled:	Is GMLC Enabled true
Destination If LocQOS Absent:	Destination If LocQOS Absent udm
Switch To Udm On Failure:	Switch To Udm On Failure ALL
Explicit Cancellation:	Explicit Cancellation false
GMLC Haccuracy:	GMLC Haccuracy 10
GMLC Vaccuracy:	GMLC Vaccuracy 10
Reporting Interval:	Reporting Interval 3600
Switch On ErrorCodes:	

codeCause ▾	Add
503 ~~ Connection Refused	Edit Delete

4.2.1.6 Configuring QoS Options

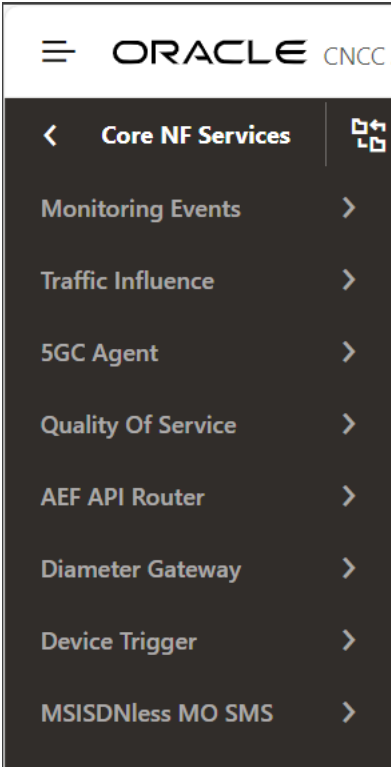
Following configurations are readonly and shall be updated only through Helm installation.

- Switch to PCRF on PCF Authorization Failure
- Enable Direct PCRF Flow

Perform the following procedure to configure QoS Options.

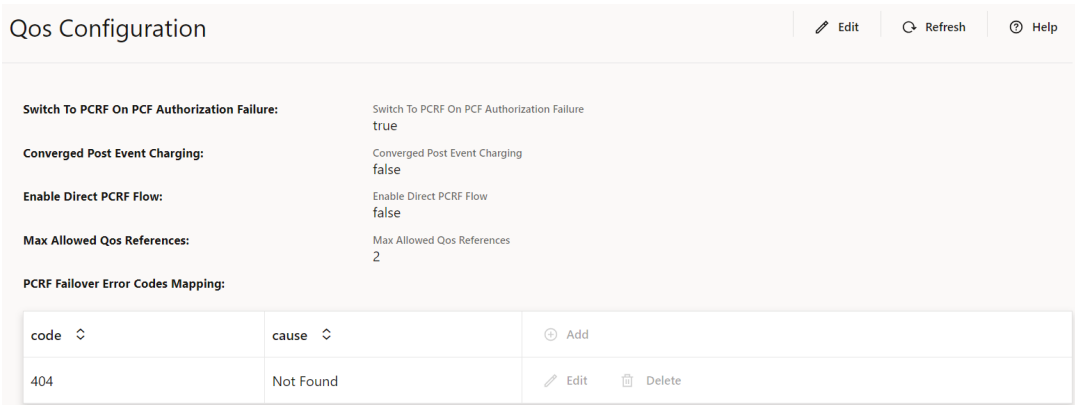
1. On the left navigation pane, click the **NEF** tab, then **Core NF Services**, and then **Quality of Service** tab.

Figure 4-22 Core NF Services



- 2. Incase if **Switch To PCRF on PCF Authorization Failure** is enabled, then PCRF Failover Error Code Mapping is configured further.

Figure 4-23 QoS Configuration



4.2.1.7 Configuring Short or Long Code for MSISDNless MO SMS

Perform the following configuration for enabling short code mapping for MSISDNless MO SMS feature.

- 1. On the left navigation pane, click the **NEF** tab, then **Core NF Services**, and then **MSISDNless MO SMS** tab.

Figure 4-24 Core NF Services

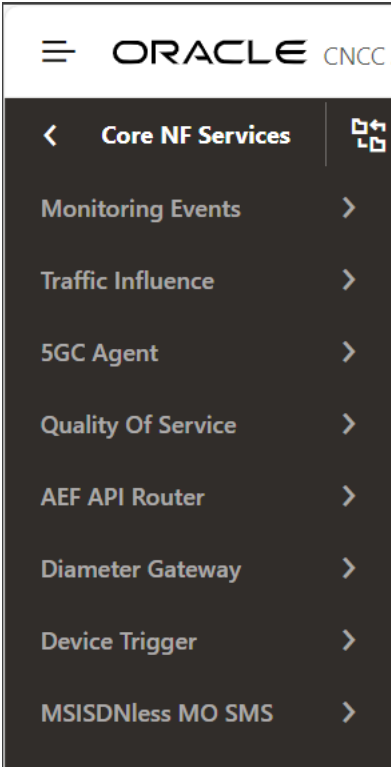


Figure 4-25 Shortcode Notification

A screenshot of the 'Shortcode Notification' configuration form. It has a light gray header with a 'Help' icon. The form contains two input fields: 'Shortcode' with the value '11811' and 'Notification URL' with the value 'http://www.example.com/mosms'. At the bottom right, there are 'Save' and 'Cancel' buttons.

- 2. Click **Shortcode Configuration**.

Figure 4-26 Shortcode Configuration

A screenshot of the 'Shortcode Configuration' table. The table has a header with a search bar 'Type to Filter', a 'Refresh' button, and an 'Add' button. The table contains two rows of data, each with a 'Shortcode', a 'Notification URL', and 'Actions' (edit and delete icons).

Shortcode	Notification URL	Actions
9876543210	http://ocnefsim-ocstub-svc-af:1010/af/notification	
31628870634	http://ocnefsim-ocstub-svc-af:1010/af/notification	

4.2.1.8 Configuring SCS Short Messaging Entity

Perform the following procedure to add AF service Id configuration applicable for Traffic Influence feature.

On the left navigation pane, click the **NEF** tab, then **Network Configuration**, and then **SCS Short Message Entity** tab.

Figure 4-27 Network Configuration

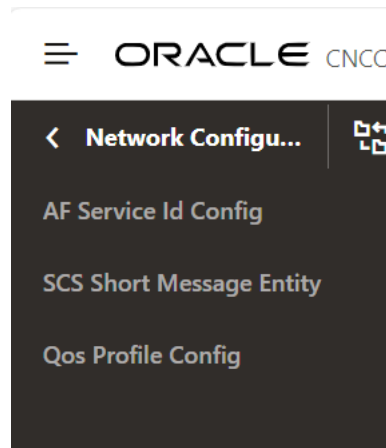


Figure 4-28 SCS Short Message Entity Configuration

SCS Short Message Entity Configuration		Edit	Refresh
Address:	Address 456789		
Type Of Number:	Type Of Number ALPHANUMERIC		
Numbering Plan Identification:	Numbering Plan Identification PRIVATE_NUMBERING_PLAN		

4.2.1.9 Configuring PLMN ID Mapping

Perform the following procedure to configure PLMN ID mapping.

On the left navigation pane, click the **NEF** tab, select **Global Options**, and then **Network Configuration** and select **PLMN Id Map Config** tab.

Figure 4-29 PLMN ID Map Configuration

PLMN Id Map Configuration Help			
Type to Filter ✕		Refresh	Add
PLMN ID	MCC	MNC	Actions
plmnlid1	234	123	✎ ✖
plmnid2	123	124	✎ ✖

Note

You can modify or delete an existing configuration by using the respective actions.

Adding PLMN ID Mapping

Perform the following procedure to add a PLMN ID mapping:

1. Click **Add**.
2. In **Create PLMN Id Map Configuration** window, enter PLMN ID, MCC, and MNC configuration.
3. Click **Save**.

4.2.1.10 Configuring TAI Mapping

Perform the following procedure to configure TAI mapping.

On the left navigation pane, click the **NEF** tab, then **Global Options**, then **Network Configuration**, and then **TAI Map Config** tab.

Figure 4-30 TAI Map Configuration

TAI Map Configuration Help				
Type to Filter ✕		Refresh	Add	
TAI ID	PLMN ID	TAC	NID	Actions
taild1	plmnlid1	12123		✎ ✖
TAIID2	plmnid2	124444		✎ ✖

Note

You can modify or delete an existing configuration by using the respective actions.

Adding TAI Mapping**Prerequisites:**

- TAI Map should be associated with the configured PLMN ID Map. All the configured PLMN ID Map are available in the drop-down, PLMN ID. Refer to [Configuring PLMN ID Mapping](#) for configuring PLMN ID map.

Perform the following procedure to add a new TAI mapping:

- Click **Add**.
- In **Create TAI Map Configuration** window, enter TAI ID, PLMN ID, TAC, and NID configuration.

Note

PLMN ID is a dynamic drop-down option based on the configuration in PLMN ID map.

Figure 4-31 Create TAI Map Configuration

Create TAI Map Configuration Help

TAI ID:

PLMN ID:

TAC:

NID:

- Click **Save**.

4.2.1.11 Configuring ECGI Mapping

Perform the following procedure to configure ECGI mapping.

On the left navigation pane, click the **NEF** tab, then **Global Options**, then **Network Configuration**, and then **ECGI Map Config** tab.

Figure 4-32 ECGI Map Config

ECGI Map Configuration Help

Type to Filter Refresh Add

ECGI ID	PLMN ID	Eutra Cell Id	NID	Actions
ecgild1	plmnlid1	23		
ecgid3	plmnid2	123	123	

Note

You can modify or delete an existing config by using the respective actions.

Adding ECGI Mapping

Prerequisites:

- PLMN ID map should be configured with appropriate values that need to be linked to ECGI MAP (dropdown option). Refer to [Configuring PLMN ID Mapping](#) for configuring PLMN ID map.

Perform the following procedure to add a new ECGI mapping:

1. Click **Add**
2. In **Create ECGI Map Configuration** window, enter ECGI ID, PLMN ID, Eutra Cell Id and NID configuration.

Note

PLMN ID is a dynamic dropdown option based on the configuration in PLMN ID map.

Figure 4-33 Create ECGI Map Configuration

Create ECGI Map Configuration

ECGI ID:

ECGI ID
ecgi2

PLMN ID:

PLMN ID
plmnId1

Eutra Cell Id:

Eutra Cell Id
24444

Required

NID:

NID

Save

Cancel

3. Click **Save**

4.2.1.12 Configuring NCGI Mapping

Perform the following procedure to configure NCGI mapping.

On the left navigation pane, click the **NEF** tab, then **Global Options**, then **Network Configuration**, and then **NCGI Map Config** tab.

Figure 4-34 NCGI Map Config

NCGI Map Configuration

Type to Filter

Refresh

Add

NCGI ID	PLMN ID	NR Cell Id	NID	Actions
ncgild1	plmnId1	3		<div><div></div><div></div></div>
233	plmnId1	4		<div><div></div><div></div></div>

Note

You can modify or delete an existing config by using the respective actions.

Adding NCGI Mapping

Prerequisites:

- PLMN ID map should be configured with appropriate values that need to be linked to NCGI MAP (dropdown option). Refer to [Configuring PLMN ID Mapping](#) for configuring PLMN ID map.

Perform the following procedure to add a new NCGI mapping:

- 1. Click **Add**
- 2. In **Create NCGI Map Configuration** window, enter NCGI ID, PLMN ID, NR Cell Id and NID configuration.

Note

PLMN ID is a dynamic dropdown option based on the configuration in PLMN ID map.

Figure 4-35 Create NCGI Map Configuration

Create NCGI Map Configuration

Help

NCGI ID:

NCGI ID
ncgi2

PLMN ID:

PLMN ID
plmnlid1

NR Cell Id:

NR Cell Id
34556

NID:

NID

- 3. Click **Save**

4.2.1.13 Configuring GNBID Mapping

Perform the following procedure to configure GNBID mapping.

On the left navigation pane, click the **NEF** tab, then **Global Options**, then **Network Configuration**, and then **GNBID Map Config** tab.

Figure 4-36 GNBID Map Config

GNBID Map Configuration

Help

Type to Filter

Refresh

Add

GNB Id	Bit Length	GNB Value	Actions
gnbid2	32	234444	<div><div></div><div></div></div>
gNbld1	23	134544	<div><div></div><div></div></div>

Note

You can modify or delete an existing config by using the respective actions.

Adding GNBID Mapping

Perform the following procedure to add a new GNBID mapping:

- 1. Click **Add**
- 2. In **Create GNBID Map Configuration** window, enter GNBID, Bit Length and GNB value.

Figure 4-37 Create GNBID Map Configuration

Create GNBID Map Configuration

GNB Id:

GNB Id
gnbid3

Bit Length:

Bit Length
22

GNB Value:

GNB Value
14455

Save

Cancel

- 3. Click **Save**

4.2.1.14 Configuring GlobalRANNodeID Mapping

Perform the following procedure to configure GlobalRANNodeID mapping.

On the left navigation pane, click the **NEF** tab, then **Global Options**, then **Network Configuration**, and then **GlobalRANNodeID Map Config** tab.

Figure 4-38 GlobalRANNodeID Map Config

GlobalRANNodeID Map Configuration

Type to Filter

Refresh

Add

RANNode Id	PLMN ID	N3IWF ID	GNBID	Actions
ranNodeid1	plmnlid1		gNbld1	<div><div></div><div></div></div>
ranNodeid2	plmnlid1		gNbld1	<div><div></div><div></div></div>

Note

You can modify or delete an existing config by using the respective actions.

Adding GlobalRANNodeID Mapping

Prerequisites:

- PLMN ID map and GNBID map should be configured with appropriate values that need to be linked to GlobalRANNodeID MAP (dropdown option). Refer to [Configuring PLMN ID Mapping](#) for configuring PLMN ID map and [Configuring GNBID Mapping](#) for configuring GNBID map.

Perform the following procedure to add a new GlobalRANNodeID mapping:

1. Click **Add**
2. In **Create GlobalRANNodeID Map Configuration** window, enter RANNode Id, PLMN ID, N3IWF ID, GNBID ,NGENB ID, WAGF ID, TNGF ID , ENB ID and NID configuration.

Note

PLMN ID and GNBID is a dynamic dropdown option based on the configuration.

Figure 4-39 Create GlobalRANNodeID Map Configuration

RANNode Id:	RANNode Id rannodeid2
PLMN ID:	PLMN ID plmnid1
N3IWF ID:	N3IWF ID
GNBID:	GNBID gnbid2
NGENB ID:	NGENB ID
WAGF ID:	WAGF ID
TNGF ID:	TNGF ID
NID:	NID 134555
ENB ID:	ENB ID

Save Cancel

3. Click **Save**

4.2.1.15 Configuring GeoZoneIdToSpatialValidity Mapping

Perform the following procedure to configure GeoZoneIdToSpatialValidity mapping applicable for Traffic Influence feature.

On the left navigation pane, click the **NEF** tab, then **Global Options**, then **Network Configuration**, and then **GeoZoneIdToSpatialValidity Config** tab.

Figure 4-40 GeoZoneIdToSpatialValidity Map Config

GeoZoneIdToSpatialValidity Map Configuration Help

Type to Filter ✕ Refresh Add

Zone ^	PRAID ^	AdditionalPRAID ^	PresenceState ^	Actions
zone1				✎ ✖
zone2			UNKNOWN	✎ ✖

Note

You can modify or delete an existing config by using the respective actions.

Adding GeoZoneIdToSpatialValidity Mapping**Prerequisites:**

- TAI map, ECGI map, NCGI map and GlobalRANNodeID map should be configured with appropriate values that need to be linked to GeoZoneIdToSpatialValidity MAP (dropdown option) attributes which are TrackingArea List, ECGI List, NCGI List, GlobalRANNodeID List, globaleNBID List respectively. Refer to:
 - [Configuring TAI Mapping](#) for configuring TAI map
 - [Configuring ECGI Mapping](#) for configuring ECGI map
 - [Configuring NCGI Mapping](#) for configuring NCGI map
 - [Configuring GlobalRANNodeID Mapping](#) for configuring GlobalRANNodeID map

Perform the following procedure to add a new GeoZoneIdToSpatialValidity mapping:

- Click **Add**
- In **Create GeoZoneIdToSpatialValidity Map Configuration** window, enter appropriate values for Zone, PRAID, AdditionalPRAID, PresenceState, TrackingArea List, ECGI List, NCGI List, GlobalRANNodeID List and globaleNBID List.

Note

TrackingArea List, ECGI List, NCGI List, GlobalRANNodeID List, and globaleNBID List are dynamic dropdown multi-select option based on the configuration.

Figure 4-41 Create GeoZoneIdToSpatialValidity Map Configuration
Figure 4-42 Create GeoZoneIdToSpatialValidity Map Configuration
3. Click **Save**

4.2.1.16 Viewing Global System Config

Perform the following procedure to view Global System Config.

On the left navigation pane, click the **NEF** tab, then **Global Options**, and then **System Config**.

The following information appears:

- Current Site details and peer site configurations are displayed, in case if Geo Redundancy is enabled.
- List of features that are enabled or disabled in the current deployment are shown.
- If GR is enabled, notification flow handling is shown.

Figure 4-43 System Configuration

System Configuration

Instance Id:	Instance Id 9faf1bbc-6e4a-4454-a507-aef01a101a27
Site Id:	Site Id 9faf1bbc-6e4a-4454-a507-aef01a101a27

▼

Enabled Features

Converged SCEF + NEF:	Converged SCEF + NEF true
Device Trigger:	Device Trigger true
GMLC:	GMLC false
Monitoring Events:	Monitoring Events true
MSISDNless MO SMS:	MSISDNless MO SMS true
Traffic Influence:	Traffic Influence true
Quality Of Service:	Quality Of Service true

▼

Geo Redundancy

Status:	Status DISABLED
---------	--------------------

▼

Notification Flow Checks

Check Site Status:	Check Site Status
Check DB Replication Status:	Check DB Replication Status

4.2.1.17 Configuring Gateway

4.2.1.17.1 Configuring Routes

Perform the following configurations for enabling SBI routing.

Peer Configuration

1. On the left navigation pane, click the **NEF** tab, then **Common NF Services**, and then **5gc Core Egress Gateway**.
2. Select **Peer Configuration** and click **Add** to create a new entry for Peer Configuration.
3. Click **Save** to complete the configuration.

Figure 4-44 Peer Configuration

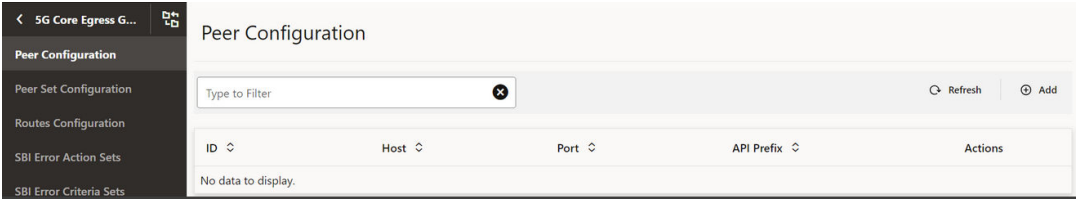
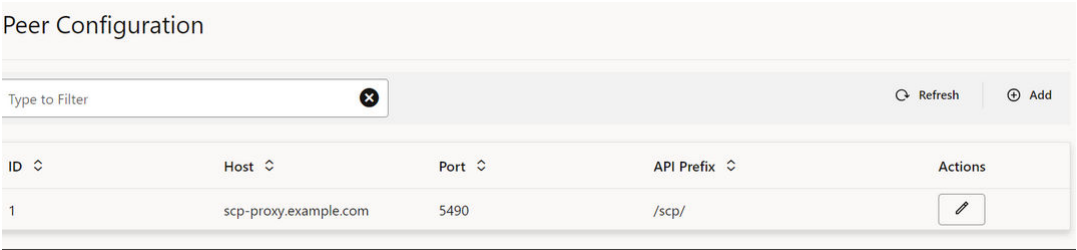


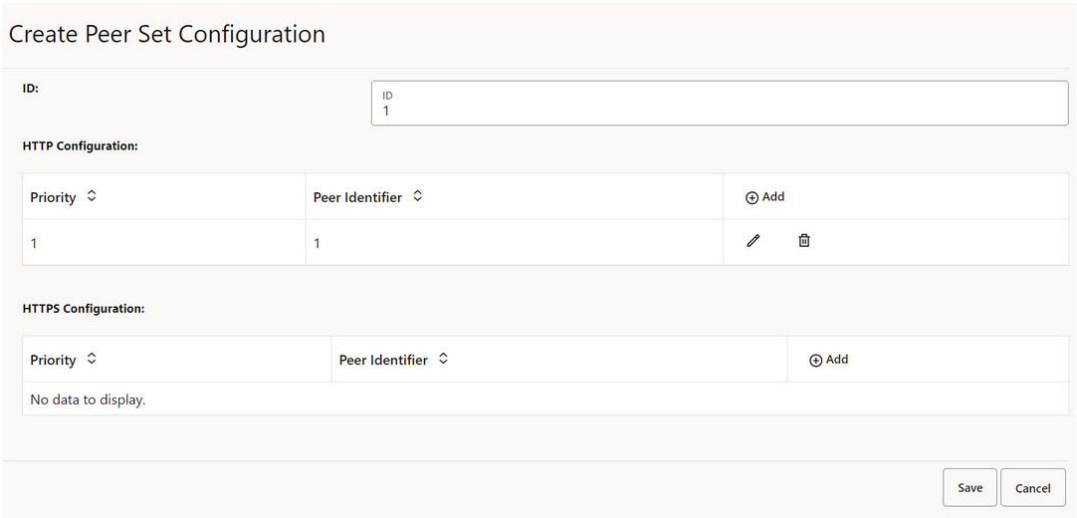
Figure 4-45 Peer Configuration - Edit



Peer Set Configuration

- 1. On the left navigation pane, click the **NEF** tab, then **Common NF Services**, and then **5gc Core Egress Gateway**.
- 2. Select **Peer Set Configuration** and click **Add** to create a new entry for Peer Set Configuration. Refer to the Peer ID configured in previous step.

Figure 4-46 Create Peer Set Configuration



- 3. Click **Save** to complete the configuration.

Route Configuration

- 1. On the left navigation pane, click the **NEF** tab, then **Common NF Services**, and then **5gc Core Egress Gateway**.
- 2. Select **Routes Configuration** and click **Add** to add a new Route configuration.

Figure 4-47 Route Configuration

ID:

1

URI:

URI: http://scp-proxy-example.com

Order:

Order 1

metadata

httpsTargetOnly:

httpRuriOnly:

stblRoutingEnabled:

configurableErrorCodes

Enabled:

errorScenarios:

exceptionType

errorProfileName

Add

No data to display.

Add

Remove

predicates:

Name

Add

No data to display.

SBI Error Action Sets

1.

On the left navigation pane, click the **NEF** tab, then **Common NF Services**, and then **5gc Core Egress Gateway**.
2.

Select **SBI Error Action Sets**.

Figure 4-48 SBI Error Action Sets

SBI Error Action Sets

Type to Filter

Refresh

Add

ID	Action	Attempts	Actions
No data to display.			

SBI Error Criteria Sets

1.

On the left navigation pane, click the **NEF** tab, then **Common NF Services**, and then **5gc Core Egress Gateway**.
2.

Select **SBI Error Criteria Sets**.

Figure 4-49 SBI Error Criteria Sets

SBI Error Criteria Sets

Type to Filter

Refresh

Add

ID	Method	Exceptions	Actions
No data to display.			

4.2.1.18 Viewing cnDBTier APIs in CNC Console

Perform the following procedure to view cnDBTier APIs in CNC Console.

Backup List

On the left navigation pane, click the **NEF** tab, then **cnDBTier**, and then **BackupList**.

Figure 4-50 Backup List

Backup List Refresh

Site Name:

Site Name
bangaloresiteone

Backup Details:

Backup Id	Backup Size (bytes)	Creation TimeStamp
215240007	223744	2024-02-15 00:07:50.0
222240007	2933248	2024-02-22 00:07:49.0

cnDBTier Version

On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **cnDBTier Version**.

Figure 4-51 cnDBTier Version

cnDBTier Version Refresh

cnDBTier Version:

cnDBTier Version
24.1.0-beta.3

NDB Version:

NDB Version
ndb-8.0.35

Database Stats

On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **Database Statistics Reports**.

Figure 4-52 Database Statistics Reports

Database Statistics Report Refresh

Database Count:

Database Count
7

Database Tables Count:

Database Name	Table Count
hbasepica_info	1
oncsdbmv	92
oncsdbmv	92
backup_info	3
monocommonconfigmv	2
replication_info	7
monocommonconfigmv	2

Database Table Rows Count:

Database Name	
hbasepica_info	View
oncsdbmv	View
oncsdbmv	View
backup_info	View
monocommonconfigmv	View
replication_info	View

GeoReplication Status

On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **GeoReplication Status**.

Figure 4-53 GeoReplication Status

Georeplication Status

Type to Filter

Refresh

Local Site Name	Remote Site Name	Replication Status	Seconds Behind Remote Site	Actions
bangaloresiteone	bangaloresitetwo	UP	0	

HeartBeat Status

On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **HeartBeat Status**.

Figure 4-54 HeartBeat Status

Replication HeartBeat Status

Refresh

Site Name:

Site Name
bangaloresiteone

Heartbeat Details:

Remote Site Name	Heartbeat Status	Heartbeat Lag	Replication Channel Group Id	
bangaloresitetwo	SUCCESS	0	1	

Local Cluster Status

On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **Local Cluster Status**.

Figure 4-55 Local Cluster Status

Local Cluster Status

Refresh

Site Name:

Site Name
bangaloresiteone

Cluster Status:

Cluster Status
UP

On Demand Backup

On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **On Demand Backup**.

Figure 4-56 On Demand Backup

On Demand Backup

Edit
Refresh

Site Name:

Site Name

bangaloresiteone

DR Status:

DR Status

NONE

Backup Id:

Backup Id

222240007

Backup Status:

Backup Status

COMPLETED

Remote Transfer Status:

Remote Transfer Status

NONE

Initiate Backup:

Initiate Backup

false

Replication Health Status

- On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **cnDBTier Health**.
- Click **Replication Health Status**.

Figure 4-57 Replication Health Status

Replication Health Status

Refresh

Local Site Name:

Local Site Name

bangaloresiteone

Health Status Details:

Service Name	Service Status	DB Connection Status	Overall Replication Service Health
mysql-cluster-bangaloresiteone-bangaloresitetwo-replication-svc.nef-cndbtier-1	UP	UP	UP

Monitor Health Status

- On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **cnDBTier Health**.
- Click **Monitor Health Status**.

Figure 4-58 Monitor Health Status

Monitor Health Status

Service Name:	Service Name mysql-cluster-db-monitor-svc
DB Connection Status:	DB Connection Status UP
Metric Scrape Status:	Metric Scrape Status UP
Overall Monitor Service Health:	Overall Monitor Service Health UP

NDB Health Status

1. On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **cnDBTier Health**.
2. Click **NDB Health Status**.

Figure 4-59 NDB Health Status

NDB Health Status Refresh

Local Site Name: Local Site Name
bangaloresiteone

NDB Health Status Details:

Service Name	Service Status	PVC Health Status
ndbmgmd-0	UP	UP
ndbmgmd-1	UP	UP
ndbmtd-0	UP	UP
ndbmtd-1	UP	UP
ndbappmysqld-0	UP	NA
ndbappmysqld-1	UP	NA
ndbmysqld-0	UP	UP
ndbmysqld-1	UP	UP

Backup Manager Health Status

1. On the left navigation pane, click the **NEF** tab, then, **cnDBTier**, and then **cnDBTier Health**.
2. Click **Backup Manager Health Status**.

Figure 4-60 Backup Manager Health Status

Backup Manager Health Status		Refresh
Service Name:	Service Name mysql-cluster-db-backup-manager-svc	
Service Status:	Service Status UP	
DB Connection Status:	DB Connection Status UP	
Overall Backup Manager Service Health:	Overall Backup Manager Service Health UP	
Backup Executor Health Status:		
Node Id	DB Connection Status	
2	UP	
1	UP	

4.2.2 Configuring CAPIF Features

This section provides information about enabling the following features of CAPIF:

Note

You must log in to the CNC Console while performing the procedures described in the subsequent subsections.

- Configuring log level for various services
- Configuring Invoker Access Token
- Configuring discovery group
- Configuring invoker pre-provisioning

4.2.2.1 Configuring Log Level for Services

Perform the following procedure to update log level for various services.

1. On the left navigation pane, click the **CAPIF** tab, then **General Options**, and then **Logging Level Options** tab.
The current list of services in CAPIF deployment and the corresponding log levels appears.

Figure 4-61 Logging Level Options

ORACLE

CNCC 24.2.0-alpha.5

Cluster1.CAPIF.capiF

About

Sign Out

General Options

Logging Level Options

Discovery Group

Invoker Pre-Provisioning

Logging Level Options

Edit

Refresh

Log Level List:

Service	Application Log Level	
AFMgr	INFO	<div><div></div>View</div>
EventMgr	INFO	<div><div></div>View</div>
ApiMgr	INFO	<div><div></div>View</div>

2. Click **Edit** to change log level of a specific service type.
The **Edit Logging Level Options** page appears.

3. Select the required **Service Type** and the corresponding **Application Log Level**.

Figure 4-62 Edit Logging Level Options

Edit Logging Level Options

Service Type:

Application Log Level:

Package Log Level:

Package	Log Level	
root	WARN	Edit

[Save](#) [Cancel](#)

4. Click **Edit** to change the **Package** and **Log Level** of the corresponding Service Type.

Figure 4-63 Edit Package Log Level

Service Type
AFMgr

Edit Package Log Level

Package:

Log Level:

[Save](#) [Cancel](#)

5. Click **Save**.

4.2.2.2 Configuring Invoker Access Token

Perform the following procedure to update the expiry time for Invoker Access Token.

1. On the left navigation pane, click the **CAPIF** tab, then **Core Capif Services**, and then **AF Manager** tab.
2. On the left navigation pane, click **Access Token Config**.

Figure 4-64 Invoker Access Token Configuration

Invoker Access Token Configuration

expiryTime: 3600

Edit Refresh

- Click **Edit** to change the expiryTime value.
The **Edit Invoker Access Token Configuration** page appears.

Figure 4-65 Edit Invoker Access Token Configuration

Edit Invoker Access Token Configuration

expiryTime:

Enter a number between 0 and 86400.

Save Cancel

- Click **Save**.

4.2.2.3 Configuring Discovery Group

Perform the following procedure to add Discovery Group.

Note

You can delete the existing config by using the **Delete** button on the config.

- On the left navigation pane, click the **CAPIF** tab and then **General Option** tab.
- To view the Discovery group, on the left navigation pane, click **Discovery Group**.

Figure 4-66 Discovery Groups

ORACLE CNCC 24.2.0-alpha.5 Cluster1.CAPIF.capiif About Sign Out

General Options Logging Level Options Discovery Group Invoker Pre-Provisioning

Discovery Groups

Type to Filter Refresh Add

Group ID	Actions
group001	Delete Edit

- Click **Add** to create new discovery group.
- Provide the Discovery Group ID name and select `nef` services from the drop down list.

Figure 4-67 Adding Discovery Group

Group ID:

NEF Services:

Group ID
Group1

NEF Services
Please enter value(s) here

RegSec::3gpp-as-session-with-qos

RegSec::3gpp-monitoring-event

RegSec::3gpp-traffic-influence

5. Click **Save**.

Figure 4-68 Discovery Group Added

General Options

Logging Level Options

Discovery Group

Invoker Pre-Provisioning

Discovery Groups

Type to Filter

Refresh

Add

Group ID	Actions
group001	<div></div> <div></div>
Group1	<div></div> <div></div>

4.2.2.4 Configuring Invoker Pre-provisioning

Perform the following procedure to add or update or delete Invoker Pre-provisioning.

- 1. On the left navigation pane, click the **CAPIF** tab and then **General Option** tab.
- 2. To add or update or delete API Invoker, on the left navigation pane, click **Invoker Pre-Provisioning**.

Figure 4-69 Invoker Pre-Provisioning

General Options

Logging Level Options

Discovery Group

Invoker Pre-Provisioning

Invoker Pre-Provisioning

Type to Filter

Refresh


Add

Invoker Name	Actions
afID1000	<div>Invoker Name</div> <div></div> <div></div> <div></div>

Figure 4-70 Edit Invoker Pre-Provisioning

<

General Options



Logging Level Options

Discovery Group


Invoker Pre-Provisioning

Invoker Name:

Invoker Name
Invoker1


Valid From:

5/17/2024, 12:00 AM




Valid To:

5/24/2024, 12:00 AM



Group ID:

Group ID
Group1



Access Token:

Access Token

Onboarding Uri:

Onboarding Uri
Value would be generated upon saving this page.

Save

Cancel

3. Click **Save**.

Figure 4-71 Invoker Pre-Provisioning

Invoker Pre-Provisioning

✕

🔄 Refresh ➕ Add

Invoker Name	Actions
Invoker1	✎ 🗑 📄
aIID1000	✎ 🗑 📄

4. To open the invoker and copy the Access token and Onboarding URI details, click **Edit** action icon.

Figure 4-72 Edit Invoker Pre-Provisioning

<

General Options

Logging Level Options

Discovery Group

Inverter Pre-Provisioning

Invoker Name:

Invoker1

Valid From:

5/17/2024, 12:00 AM

📅

Valid To:

5/24/2024, 12:00 AM

📅

Group ID:

Group1

▼

Access Token:

Access token
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZSI6ImF1dXludm9rZXIiwiaWF0IjE3NzYyMzY1OTQwMDAwMDA6MTZlLnRlLFJmZW5kENDIzY05uJHlwYyV1ZDZkZDY2LzY5InsiNTY5aWkiPmFwaUludm9rZXUicmVmcm90c2x0Npb25ibmclLCAprfXQIQIE3MTUA0ODQyNDAdm14CDBMTcnQ0T0AwMHd_YKtois-SV-eDXTPncoSSHvswBEu483-cczhC9wZcu4avafyBt_BjgNGhtpckGMaj93sozeR4Dwe6SvCygzBVm-pIRREXX_uog5HvpJF5_CrySY0zmymONKhq934wbQ30miID4P9nnjTLRO_J0BL_IIJ327sdMR3dcyOLWehJd3SczkpKMts2s-Cg-ZJL-O-xid8ETEby43Z1eyn4QbvuytL_xd8MIQ_FmJvaAJjsaa73tyEAQISDHCDCLCOkay8DA3n9lqe98_JoSiOqhnc3d9mgCGBNYPgm7rmkTejoHL_P-fj-AVNhHmeSrvnd02RuAXI.Rw

Onboarding URL:

Onboarding URI
<http://occapit-ux-ingress-gateway:80/apiRoot/api-inverter-management/v1/onboardedinvokers>

Save

Cancel

5

Configuring NEF

This section provides information about configuring Oracle Communications Cloud Native Core Network Exposure Function (NEF).

NEF offers the Custom Value files to customize CAPIF and NEF deployment.

The NEF deployment can be customized by overriding the default values of various configurable parameters in the custom value files. For more information about downloading and customizing the custom value files for CAPIF and NEF, see *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

6

NEF Metrics

This chapter includes information about Metrics for Oracle Communications Cloud Native Core, Network Exposure Function (NEF).

6.1 NEF OCI Metrics Dashboard

To upload the `ocnef_oci_metric_dashboard_24.2.2.zip` file on OCI Logging Analytics Dashboard Service, see *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*. OCI does not support Grafana, OCI uses the Logging Analytics Dashboard Service for visualizing the metrics and logs.

6.2 Dimension Description

The following table provides information on Dimension of Metrics.

Table 6-1 NEF Dimension Description

Dimension	Description	Values
Method	Type of security method	string
Status	Status	string
Service Name	The complete name of current service	string
query_type	Type of query	string
Svc_Type	Type of the service	enum
ApiName	The name of the API	string
Cause	Reason for failure	string
error	Error message	boolean
Host	Host address	IP address
Port	Port number	port
Direction	Direction of the service request	string
Event	The event that occurred on request completion	string
Client_type	HTTP client type The possible value is h2c.	string
DestinationHost	The request destination host address	IP address

Note

If any specific KPI is required, then create this KPI on grafana dashboard as per requirement from this metrices mentioned.

6.3 NEF Metrics

This section provides all the NEF metrics and respective dimensions.

Note

For more information about Dimensions, see [NEF Metrics](#).

Examples

The following are some of the examples of NEF metrics:

CCF Client Metrics

```
ocnef_ccf_client_publish_total{Method="POST", ServiceName="ocnef-
monitoringevents", app_kubernetes_io_instance="ocnef",
app_kubernetes_io_name="ocnef-ccfclient", container="ocnef-ccfclient",
endpoint="cnc-metrics", instance="10.233.69.246:9090",
job="ocne-infra/ocne-nf-cnc-podmonitor", namespace="nef-perf", pod="ocnef-
ocnef-ccfclient-6cc95c69dd-mjb74", pod_template_hash="6cc95c69dd"}
```

```
ocnef_ccf_client_registration{Success="No",
app_kubernetes_io_instance="ocnef", app_kubernetes_io_name="ocnef-ccfclient",
container="ocnef-ccfclient", endpoint="cnc-metrics",
instance="10.233.69.246:9090", job="ocne-infra/ocne-nf-cnc-podmonitor",
namespace="nef-perf", pod="ocnef-ocnef-ccfclient-6cc95c69dd-mjb74",
pod_template_hash="6cc95c69dd"}
```

ME Service Metrics

```
ocnef_me_5g_req_total{method="POST"} 1.0
```

```
ocnef_me_db_query_seconds_max{app="monitoringevents",method="POST"} 1.0
```

```
ocnef_5gc_agent_req_total{method="POST",from="5G_NF"} 1.0
```

```
ocnef_5gc_agent_resp_total{method="POST",status="200",to="5G_NF"} 1.0
```

6.3.1 API Router Metrics

The following metrics are applicable to API router metrics.

Table 6-2 ocnef_aef_apirouter_req

Field	Details
Description	Count of request messages received by the API Router service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method API Name

Table 6-2 (Cont.) ocnef_aef_apirouter_req

Field	Details
Example	

Table 6-3 ocnef_apirouter_resp

Field	Details
Description	Count of response messages sent back by the API Router service.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Status • API Name
Example	

Table 6-4 ocnef.aef.apirouter.latency

Field	Details
Description	The amount of time taken for processing of the request at the API Router service.
Type	Histogram
Dimension	<ul style="list-style-type: none"> • API Name • Method • error(boolean)
Example	

Table 6-5 ocnef.aef.apirouter.backend.connection.failure

Field	Details
Description	The number of connection creation failure with the backend NEF microservice.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Host • Port
Example	

Table 6-6 ocnef.aef.apirouter.backend.outgoing.connections

Field	Details
Description	The number of active connections with backend NEF microservice.
Type	Gauge
Dimension	<ul style="list-style-type: none"> • Host • Direction
Example	

Table 6-7 ocnef.aef.apirouter.backend.request.stat

Field	Details
Description	The number of events occurred when a request is sent to backend NEF microservices.
Type	Counter
Dimension	<ul style="list-style-type: none"> event client_type DestinationHost
Example	

Table 6-8 ocnef.aef.apirouter.backend.response.stat

Field	Details
Description	The number of events occurred when a response is received at backend NEF microservices.
Type	Counter
Dimension	<ul style="list-style-type: none"> event client_type DestinationHost
Example	

Table 6-9 ocnef.aef.apirouter.backend.latency

Field	Details
Description	The amount of time taken in processing of the request at backend nef microservices.
Type	Histogram
Dimension	<ul style="list-style-type: none"> Host Method
Example	

6.3.2 CCF Client Metrics

The following metrics are applicable to CCF client metrics.

Table 6-10 ocnef_ccf_client_publish_total

Field	Details
Description	Number of CCF client requests published.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method ServiceName
Example	ocnef_ccf_client_publish_total{Method="POST", ServiceName="nefsiteone-devicetrigger"}

Table 6-11 ocnef_ccf_client_registration

Field	Details
Description	CCF client registration status
Type	Gauge
Dimension	Success
Example	ocnef_ccf_client_registration{Success="Yes"}

Table 6-12 ocnef.ccf.client.service.failure

Field	Details
Description	CCF client service failure count
Type	Counter
Dimension	<ul style="list-style-type: none"> ServiceName Status
Example	ocnef_ccf_client_service_failure{ServiceName="nfsitetwo-msisdnlessmosms", Status="Down"}

6.3.3 ME Service Metrics

The following metrics are applicable to ME service metrics.

Table 6-13 ocnef_me_af_req_total

Field	Details
Description	Count of subscriptions received at the ME service.
Type	Counter
Dimension	Method
Example	

Table 6-14 ocnef_me_af_resp_total

Field	Details
Description	Count of subscription responses sent by the ME service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-15 ocnef_me_5g_req

Field	Details
Description	Count of notifications received from 5G core at the ME service.
Type	Counter
Dimension	Method
Example	

Table 6-16 ocnef_me_5g_resp

Field	Details
Description	Count of notification responses sent by the ME service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-17 ocnef_me_5g_ue_r_ignored_req

Field	Details
Description	Notifications of UNREACHABLE and REGULATORY_ONLY, which are ignored at NEF and forwarded to AF.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-18 ocnef_me_af_latency

Field	Details
Description	Latency for processing subscriptions at the ME service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-19 ocnef_me_5g_latency

Field	Details
Description	Latency for processing notifications at the ME service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-20 ocnef_me_db_query_duration_seconds_sum

Field	Details
Description	Time duration in seconds to process dbQuery.
Type	Counter
Dimension	query_type
Example	

Table 6-21 ocnef_me_db_query_duration_seconds_count

Field	Details
Description	Count of number of dbQuery.
Type	Counter
Dimension	query_type
Example	

Table 6-22 ocnef_me_db_query_duration_seconds_max

Field	Details
Description	Maximum duration of time in seconds allowed to process dbQuery.
Type	Counter
Dimension	query_type
Example	

Table 6-23 ocnef_me_srv_req_total

Field	Details
Description	Count of messages transmitted by the ME services.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method to Note: The value of to can be 5G_NF or AF.
Example	

Table 6-24 ocnef_me_srv_resp_total

Field	Details
Description	Count of responses received by the ME service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status from Note: The value of from can be 5G_NF or AF.
Example	

Table 6-25 ocnef_me_srv_latency_seconds

Field	Details
Description	Latency for message received at the ME service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status from Note: The value of from can be 5G_NF or AF.

Table 6-25 (Cont.) ocnef_me_srv_latency_seconds

Field	Details
Example	

Table 6-26 ocnef_me_svc_gmlc_to_udm_switchover_count

Field	Details
Description	The number of requests sent to UDM on GMLC failure.
Type	Counter
Dimension	
Example	

Table 6-27 ocnef_me_af_surviving_site_total

Field	Details
Description	The number of AF requests handled by non owner site in georedundant deployment.
Type	Counter
Dimension	<ul style="list-style-type: none"> method owner_site_id status failure_reason Note: The value of method can be POST.
Example	

Table 6-28 ocnef_me_5g_surviving_site_total

Field	Details
Description	The number of notifications handled by non owner site in georedundant deployment.
Type	Counter
Dimension	<ul style="list-style-type: none"> method owner_site_id status failure_reason Note: The value of method can be POST.
Example	

Table 6-29 ocnef_all_site_status

Field	Details
Description	The status of the NEF sites in georedundant deployment.
Type	Gauge
Dimension	site_id
Example	

Table 6-30 ocnef_db_replication_status

Field	Details
Description	The DB replication status of the NEF sites in georedundant deployment.
Type	Gauge
Dimension	status
Example	

6.3.4 QoS Service Metrics

The following metrics are applicable to QoS Service metrics.

Table 6-31 ocnef_qos_af_req

Field	Details
Description	Count of QoS messages received from AF in NEF
Type	Counter
Dimension	Method
Example	

Table 6-32 ocnef_qos_af_resp

Field	Details
Description	Count of response sent from NEF for the received QoS session messages
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-33 ocnef_qos_5g_req

Field	Details
Description	Count of notification requests received from Fivegc Agent
Type	Counter
Dimension	<ul style="list-style-type: none"> Method From
Example	

Table 6-34 ocnef_qos_5g_resp

Field	Details
Description	Count of responses sent to Fivegc Agent
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-35 ocnef_qos_af_latency

Field	Details
Description	Latency for processing the messages received from AF
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Type
Example	

Table 6-36 ocnef_qos_5g_latency

Field	Details
Description	Latency for processing messages received from Fivegc Agent
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Type
Example	

Table 6-37 ocnef_qos_srv_req

Field	Details
Description	Count of QoS requests sent from QoS service towards Fivegc Agent / AF
Type	Counter
Dimension	<ul style="list-style-type: none"> • Event • To • Destination
Example	

Table 6-38 ocnef_qos_srv_resp

Field	Details
Description	Count of response received in QoS service
Type	Counter
Dimension	<ul style="list-style-type: none"> • Event • From • Destination
Example	

Table 6-39 ocnef_qos_srv_latency

Field	Details
Description	Latency duration of a request and response
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Destination • Time

Table 6-39 (Cont.) ocnef_qos_srv_latency

Field	Details
Example	

Table 6-40 ocnef_qos_db_latency

Field	Details
Description	Latency duration for a DB Query
Type	Counter
Dimension	<ul style="list-style-type: none"> Query_type Time
Example	

Table 6-41 ocnef_qos_af_surviving_site_total

Field	Details
Description	The number notifications handled by the non owner site in georedundant deployment
Type	Counter
Dimension	<ul style="list-style-type: none"> method owner_site_id status failure_reason <p>Note: The value of method can be POST.</p>
Example	

Table 6-42 ocnef_qos_5g_surviving_site_total

Field	Details
Description	The number of notifications handled by non owner site in georedundant deployment
Type	Counter
Dimension	<ul style="list-style-type: none"> method owner_site_id status failure_reason <p>Note: The value of method can be POST.</p>
Example	

Table 6-43 ocnef_chf_qos_req_total

Field	Details
Description	Count of the requests to CHF for QOS invocation or notification
Type	Counter
Dimensions	<ul style="list-style-type: none"> method direction

Table 6-43 (Cont.) ocnef_chf_qos_req_total

Field	Details
Examples	ocnef_chf_qos_req_total{application="ocnef",direction="INVOCATION"}

Table 6-44 ocnef_chf_qos_resp_total

Field	Details
Description	Count of the response to CHF for QoS invocation or notification
Type	Counter
Dimensions	<ul style="list-style-type: none"> method direction status
Examples	ocnef_chf_qos_resp_total{application="ocnef",direction="NOTIFICATION"}

Table 6-45 ocnef_chf_qos_latency

Field	Details
Description	Latency for processing the messages received from AF.
Type	Histogram
Dimensions	<ul style="list-style-type: none"> method direction
Examples	ocnef_chf_qos_latency_seconds{application="ocnef",direction="INVOCATION",eng_version="",method="DELETE"}

Table 6-46 ocnef_qos_srv_req_total

Field	Details
Description	Count of QoS requests sent from QoS service towards Fivegc Agent or AF
Type	Counter
Dimensions	<ul style="list-style-type: none"> event to destination
Examples	ocnef_qos_srv_req{to="5G_NF_CHF"}

Table 6-47 ocnef_qos_srv_resp_total

Field	Details
Description	Count of response received in QoS service.
Type	Counter

Table 6-47 (Cont.) ocnef_qos_srv_resp_total

Field	Details
Dimensions	<ul style="list-style-type: none"> event from destination
Examples	ocnef_qos_srv_resp{to="5G_NF_CHF"}

6.3.5 5GC Agent Service Metrics

The following metrics are applicable to 5GC Agent Service metrics.

Table 6-48 ocnef_5gc_agent_req_total

Field	Details
Description	Count of messages received at Fivegc Agent service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method from <p>Note: The value of from can be 5G_NF or NEF_SVC.</p>
Example	

Table 6-49 ocnef_5gc_agent_resp_total

Field	Details
Description	Count of message responses sent by the Fivegc Agent service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status to <p>Note: The value of to can be 5G_NF or NEF_SVC.</p>
Example	

Table 6-50 ocnef_5gc_agent_latency_seconds

Field	Details
Description	Latency for processing messages at the Fivegc Agent service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status from <p>Note: The value of from can be 5G_NF or NEF_SVC.</p>
Example	

Table 6-51 ocnef_5gc_agent_srv_req_total

Field	Details
Description	Count of messages transmitted by the Fivegc Agent service.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • to • Communication Model Note: The value of to can be 5G_NF or NEF_SVC.
Example	

Table 6-52 ocnef_5gc_agent_srv_resp_total

Field	Details
Description	Count of responses received by the Fivegc Agent service.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Status • from • Communication Model Note: The value of from can be 5G_NF or NEF_SVC.
Example	

Table 6-53 ocnef_5gc_agent_srv_latency_seconds

Field	Details
Description	Latency for message received at the Fivegc Agent service.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Status • from • Communication Model Note: The value of from can be 5G_NF or NEF_SVC.
Example	

Table 6-54 ocnef_translation_count

Field	Details
Description	Count of NF message translation done by Fivegc Agent.
Type	Counter
Dimension	<ul style="list-style-type: none"> • translationType • Communication Model
Example	

Table 6-55 ocnef_translation_failure_count

Field	Details
Description	Count of failed NF message translation done by Fivegc Agent.
Type	Counter
Dimension	<ul style="list-style-type: none"> translationType Communication Model
Example	

Table 6-56 ocnef_5gc_invalid_config

Field	Details
Description	The number of invalid configurations on 5GCAgent service.
Type	Counter
Dimension	
Example	

6.3.6 Expiry Auditor Service Metrics

The following metrics are applicable to Expiry Auditor Service metrics.

Table 6-57 ocnef_exp_audit_expired_records

Field	Details
Description	The number of records expired and deleted.
Type	Counter
Dimension	<ul style="list-style-type: none"> serviceld Note: The value of serviceld can be MonitoringEventSvc .
Example	

Table 6-58 ocnef_exp_audit_request_records

Field	Details
Description	The number of requests for addition and deletion of records.
Type	Counter
Dimension	<ul style="list-style-type: none"> serviceld Note: The value of serviceld can be MonitoringEventSvc .
Example	

Table 6-59 ocnef_exp_audit_response_records

Field	Details
Description	The number of records added or deleted.
Type	Counter
Dimension	<ul style="list-style-type: none"> serviceld Note: The value of serviceld can be MonitoringEventSvc .

Table 6-59 (Cont.) ocnef_exp_audit_response_records

Field	Details
Example	

6.3.7 Traffic Influence Metrics

The following metrics are applicable to Traffic Influence metrics.

Table 6-60 ocnef_traffic_influence_af_req

Field	Details
Description	Count of traffic_influence messages received from AF in NEF.
Type	
Dimension	<ul style="list-style-type: none"> Method
Example	

Table 6-61 ocnef_traffic_influence_af_resp

Field	Details
Description	Count of response sent from NEF for the received traffic_influence session messages.
Type	
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-62 ocnef_traffic_influence_5g_req

Field	Details
Description	Count of notification requests received from Fivegc Agent.
Type	
Dimension	<ul style="list-style-type: none"> Method from
Example	

Table 6-63 ocnef_traffic_influence_5g_resp

Field	Details
Description	Count of responses sent to Fivegc Agent.
Type	
Dimension	<ul style="list-style-type: none"> Method Status
Example	

Table 6-64 ocnef_traffic_influence_af_latency

Field	Details
Description	Latency for processing the messages received from AF.
Type	
Dimension	<ul style="list-style-type: none"> • Method • Type
Example	

Table 6-65 ocnef_traffic_influence_5g_latency

Field	Details
Description	Latency for processing messages received from Fivegc Agent.
Type	
Dimension	<ul style="list-style-type: none"> • Method • Type
Example	

Table 6-66 ocnef_traffic_influence_srv_req

Field	Details
Description	Count of traffic influence requests sent from traffic_influence service towards Fivegc Agent / AF.
Type	
Dimension	<ul style="list-style-type: none"> • Event • To • Destination
Example	

Table 6-67 ocnef_traffic_influence_srv_resp

Field	Details
Description	Count of response received in traffic_influence service.
Type	
Dimension	<ul style="list-style-type: none"> • Event • From • Destination
Example	

Table 6-68 ocnef_traffic_influence_srv_latency

Field	Details
Description	Latency duration of a request and response.
Type	
Dimension	<ul style="list-style-type: none"> • Method • Time • Destination

Table 6-68 (Cont.) ocnef_traffic_influence_srv_latency

Field	Details
Example	

Table 6-69 ocnef_traffic_influence_db_latency

Field	Details
Description	Latency duration for a DB Query.
Type	
Dimension	<ul style="list-style-type: none"> query_type Time
Example	

6.3.8 Converged SCEF-NEF Metrics

The following metrics are applicable to Converged SCEF-NEF metrics.

Table 6-70 ocnef_diamgw_diam_req_total

Field	Details
Description	This metric records the total number of diameter inbound and outbound requests to and from the diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code message_direction
Example	ocnef_diamgw_diam_req_total{app_id="16777313"}

Table 6-71 ocnef_diamgw_diam_resp_total

Field	Details
Description	This metric records the total number of diameter response received for the inbound and outbound requests sent to and from the diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code status message_direction
Example	ocnef_diamgw_diam_resp_total{app_id="16777313"}

Table 6-72 ocnef_diamgw_http_req_total

Field	Details
Description	This metric records the total number of inbound and outbound requests to and from the diameter gateway to NEF microservices.
Type	Counter

Table 6-72 (Cont.) ocnef_diamgw_http_req_total

Field	Details
Dimension	<ul style="list-style-type: none"> app_id cmd_code message_direction
Example	ocnef_diamgw_http_req_total{app_id="16777313"}

Table 6-73 ocnef_diamgw_http_resp_total

Field	Details
Description	This metric records the total inbound and outbound responses to and from the NEF microservices to diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code status message_direction
Example	ocnef_diamgw_http_resp_total{app_id="16777313"}

Table 6-74 ocnef_diamgw_translator_request_total

Field	Details
Description	This metric captures total requests translated, translation status from diameter to http.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code translatorType translationStatus errorReason
Example	

6.3.9 Device Trigger Metrics

The following metrics are applicable to Device Trigger metrics.

Table 6-75 ocnef_dt_af_req_total

Field	Details
Description	Count of device_trigger messages received from AF in NEF.
Type	Counter
Dimension	method
Example	

Table 6-76 ocnef_dt_af_resp_total

Field	Details
Description	Count of response sent from NEF for the received device_trigger session messages.
Type	Counter
Dimension	<ul style="list-style-type: none"> method status
Example	

Table 6-77 ocnef_dt_af_latency_seconds

Field	Details
Description	Latency for processing the messages received from AF.
Type	Histogram
Dimension	<ul style="list-style-type: none"> method type
Example	

Table 6-78 ocnef_dt_srv_req_total

Field	Details
Description	Count of device trigger requests sent from device_trigger service towards Fivegc Agent / AF.
Type	Counter
Dimension	<ul style="list-style-type: none"> event to destination
Example	

Table 6-79 ocnef_dt_srv_resp_total

Field	Details
Description	Count of response received in device_trigger service.
Type	Counter
Dimension	<ul style="list-style-type: none"> event from destination
Example	

Table 6-80 ocnef_dt_srv_latency_seconds

Field	Details
Description	Latency duration of a request and response.
Type	Histogram
Dimension	<ul style="list-style-type: none"> method destination time

Table 6-80 (Cont.) ocnef_dt_srv_latency_seconds

Field	Details
Example	

Table 6-81 ocnef_svc_dt_db_latency_seconds

Field	Details
Description	Latency duration for a DB Query.
Type	Histogram
Dimension	<ul style="list-style-type: none"> query_type time
Example	

Table 6-82 ocnef_dt_diamgw_notify_req_total

Field	Details
Description	Total number of delivery report notifications requests that was received from diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> method type
Example	

Table 6-83 ocnef_dt_diamgw_notify_resp_total

Field	Details
Description	Total number of delivery report notifications responses that was sent to diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> method status
Example	

Table 6-84 ocnef_dt_diamgw_notify_latency_seconds

Field	Details
Description	Latency duration of a request and response.
Type	Histogram
Dimension	<ul style="list-style-type: none"> method destination time
Example	

Table 6-85 ocnef_diamgw_diam_req_total

Field	Details
Description	This metric captures total diameter outbound or inbound requests from/to diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> • app_id • cmd_code • direction
Example	

Table 6-86 ocnef_diamgw_diam_resp_total

Field	Details
Description	This metric captures total diameter response received for the outbound or inbound requests sent from/to diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> • app_id • cmd_code • direction • status
Example	

Table 6-87 ocnef_diamgw_http_req_total

Field	Details
Description	This metric captures total inbound/outbound requests to/from diameter gateway to NEF microservices.
Type	Counter
Dimension	<ul style="list-style-type: none"> • app_id • cmd_code • api_name • direction
Example	

Table 6-88 ocnef_diamgw_http_resp_total

Field	Details
Description	This metric captures total incoming or outgoing responses to/from NEF microservices to diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> • app_id • cmd_code • api_name • status • direction
Example	

Table 6-89 ocnef_diamgw_translator_request_total

Field	Details
Description	This metric captures total requests translated and the translation status from diameter to http.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code translatorType translationStatus errorReason
Example	

6.3.10 Diameter Gateway Metrics

The following metrics are applicable to Diameter Gateway metrics.

Table 6-90 ocnef_diamgw_diam_req_total

Field	Details
Description	This metric records the total number of diameter inbound and outbound requests to and from the diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code message_direction
Example	ocnef_diamgw_diam_req_total{app_id="16777313"}

Table 6-91 ocnef_diamgw_diam_resp_total

Field	Details
Description	This metric records the total number of diameter response received for the inbound and outbound requests sent to and from the diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code status message_direction
Example	ocnef_diamgw_diam_resp_total{app_id="16777313"}

Table 6-92 ocnef_diamgw_http_req_total

Field	Details
Description	This metric records the total number of inbound and outbound requests to and from the diameter gateway to NEF microservices.
Type	Counter

Table 6-92 (Cont.) ocnef_diamgw_http_req_total

Field	Details
Dimension	<ul style="list-style-type: none"> app_id cmd_code message_direction
Example	ocnef_diamgw_http_req_total{app_id="16777313"}

Table 6-93 ocnef_diamgw_http_resp_total

Field	Details
Description	This metric records the total inbound and outbound responses to and from the NEF microservices to diameter gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code status message_direction
Example	ocnef_diamgw_http_resp_total{app_id="16777313"}

Table 6-94 ocnef_diamgw_translator_request_total

Field	Details
Description	This metric captures total requests translated, translation status from diameter to http.
Type	Counter
Dimension	<ul style="list-style-type: none"> app_id cmd_code translatorType translationStatus errorReason
Example	ocnef_diamgw_translator_request_total{app_id="16777313"}

6.3.11 MSISDNless MO SMS Metrics

The following metrics are applicable to MSISDNless MO SMS metrics.

Table 6-95 ocnef_msisdnless_mo_sms_diamgw_notify_req_total

Field	Details
Description	Count of (Msisdnless MO SMS notification flow) requests received from DiamGW.
Type	Counter
Dimension	<ul style="list-style-type: none"> method type
Example	ocnef_msisdnless_mo_sms_diamgw_notify_req_total{method="POST"}

Table 6-96 ocnef_msisdnless_mo_sms_diamgw_notify_resp_total

Field	Details
Description	Count of (Msisdnless MO SMS notification flow) responses sent to DiamGW.
Type	Counter
Dimension	<ul style="list-style-type: none"> method status
Example	ocnef_msisdnless_mo_sms_diamgw_notify_resp_total{method="POST"}

Table 6-97 ocnef_msisdnless_mo_sms_srv_req_total

Field	Details
Description	Count of (Msisdnless MO SMS notification flow) requests sent towards Fivegc Agent or AF.
Type	Counter
Dimension	<ul style="list-style-type: none"> event to destination
Example	ocnef_msisdnless_mo_sms_srv_req_total{event="onFailure"}

Table 6-98 ocnef_msisdnless_mo_sms_srv_resp_total

Field	Details
Description	Count of (Msisdnless MO SMS notification flow) responses received from Fivegc Agent or AF.
Type	Counter
Dimension	<ul style="list-style-type: none"> event from destination
Example	ocnef_msisdnless_mo_sms_srv_resp_total{event="onFailure"}

Table 6-99 ocnef_msisdnless_mo_sms_srv_latency

Field	Details
Description	Latency duration of request and response. It typically generates a max gauge, and some counters (_sum, _count and the buckets).
Type	Histogram
Dimension	<ul style="list-style-type: none"> method destination
Example	ocnef_msisdnless_mo_sms_srv_latency_seconds_bucket[2m]

6.4 CAPIF Metrics

This section describes the CAPIF metrics and respective dimensions.

Note

For more information about Dimensions, see [NEF Metrics](#).

Examples

The following is a sample example of EG service metrics:

```
occapifapimgr_req{Method="DELETE",Svc_Type="DiscoveryGroupService"} 1.0
```

```
occapif_apimgr_resp{Cause="NONE",Method="DELETE",Status="204",Svc_Type="DiscoveryGroupService"} 1.0
```

```
occapif_afmgr_req{Method="DELETE",Svc_Type="PreProv"} 1.0
```

```
occapif_afmgr_resp{Cause="NONE",Method="DELETE",Status="204",Svc_Type="PreProv"} 1.0
```

```
occapif_eventmgr_subscription_req{Events="SERVICE_API_AVAILABLE",Method="POST"} 1.0
```

```
occapif_eventmgr_subscription_resp{Cause="NONE",Events="SERVICE_API_AVAILABLE",Method="POST",Status="201"} 1.0
```

```
occapif_eventmgr_subscription_req{Events="SERVICE_API_AVAILABLE",Method="DELETE"} 1.0
```

```
occapif_eventmgr_subscription_resp{Cause="NONE",Events="SERVICE_API_AVAILABLE",Method="DELETE",Status="204"} 1.0
```

```
occapif_eventmgr_publish_req{Events="SERVICE_API_AVAILABLE",Method="POST"} 1.0
```

```
occapif_eventmgr_publish_resp{Cause="NONE",Events="SERVICE_API_AVAILABLE",Method="POST",Status="204"} 1.0
```

```
occapif_eventmgr_notification_req{Events="SERVICE_API_AVAILABLE",Method="POST"} 3.0
```

```
occapif_eventmgr_notification_resp_total{Cause="NONE",Events="SERVICE_API_AVAILABLE",Method="POST",Status="204"} 3.0
```

6.4.1 API Manager Metrics

The following metrics are applicable to API Manager metrics.

Table 6-100 occapif_apimgr_req

Field	Details
Description	Count of request messages received by the API Manager service.
Type	Counter

Table 6-100 (Cont.) occapif_apimgr_req

Field	Details
Dimension	<ul style="list-style-type: none"> Method Svc_Type Note: The Svc_Type contains the following details: <ul style="list-style-type: none"> RegistrationService PublishService DiscoveryService DiscoveryGroupService
Example	

Table 6-101 occapif_apimgr_resp

Field	Details
Description	Count of response messages sent back by the API Manager service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Status Cause Svc_Type
Example	

6.4.2 AF Manager Metrics

The following metrics are applicable to AF Manager metrics.

Table 6-102 occapif_afmgr_req

Field	Details
Description	Count of request messages received by the AF Manager service.
Type	Counter
Dimension	<ul style="list-style-type: none"> Method Svc_Type Note: The Svc_Type contains the following details: <ul style="list-style-type: none"> PreProv Onboarding ObtainAuth SecurityContext
Example	

Table 6-103 occapif_afmgr_resp

Field	Details
Description	Count of response messages sent back by the AF Manager service.
Type	Counter

Table 6-103 (Cont.) occapif_afmgr_resp

Field	Details
Dimension	<ul style="list-style-type: none"> • Method • Status • Cause • Svc_Type
Example	

6.4.3 Event Manager Metrics

The following metrics are applicable to Event Manager metrics.

Table 6-104 occapif_eventmgr_subscription_req

Field	Details
Description	The number of EG event subscription requests received.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Event
Example	

Table 6-105 occapif_eventmgr_subscription_resp

Field	Details
Description	The number of EG event subscription responses sent.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Event • Status • Cause
Example	

Table 6-106 occapif_eventmgr_publish_req

Field	Details
Description	The number of EG event publish requests received.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Event
Example	

Table 6-107 occapif_eventmgr_publish_resp

Field	Details
Description	The number of EG event publish response received.
Type	Counter

Table 6-107 (Cont.) occapif_eventmgr_publish_resp

Field	Details
Dimension	<ul style="list-style-type: none"> • Method • Event • Status • Cause
Example	

Table 6-108 occapif_eventmgr_notification_req

Field	Details
Description	The number of CAPIF event notification requests sent.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Event
Example	

Table 6-109 occapif_eventmgr_notification_resp_total

Field	Details
Description	The number of CAPIF event notifications responses received.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Event • Status • Cause
Example	

6.5 Ingress Gateway Metrics

Ingress Gateway Metrics

Ingress Metrics Common Tags

Tags	Description	Possible Values
Method	Http method.	<ul style="list-style-type: none"> • GET • PUT • POST • DELETE • PATCH
NFType	Name of the NF Type.	For Eg: Path is /nxxx-yyy/vz/..... Where XXX(Upper Case) is NFType UNKNOWN if unable to extract NFType from the path

Tags	Description	Possible Values
NFServiceType	Name of the Service with in the NF.	For Eg: Path is /nxxx-yyy/vz/..... Where nxxx-yyy is NFServiceType UNKNOWN if unable to extract NFServiceType from the path
Host	Port of ingress gateway (Ip or fqdn).	NA
HttpVersion	Http protocol version.	<ul style="list-style-type: none"> HTTP/1.1 HTTP/2.0
Scheme	Http protocol scheme.	HTTP, HTTPS, UNKNOWN
ClientCertIdentity	Cerificate Identity of the client.	SAN=127.0.0.1,localhost CN=localhost, N/A if data is not available
Route_Path	Path predicate/Header predicate that matched the current request.	NA
InstanceIdentifier	Prefix of the pod configured in helm when there are multiple instances in same deployment.	Prefix configured in helm otherwise UNKNOWN
ErrorOriginator	This tag captures the ErrorOriginator.	ServiceProducer, Nrf, IngresGW, None
oc_ingressgateway_route_ratelimit_Status oc_ingressgateway_global_ratelimit_Status	Request accepted or dropped.	accepted, dropped
oc_ingressgateway_connection_failure_Host	Destination ip/fqdn.	NA
oc_ingressgateway_connection_failure_Port	Destination port.	NA
oc_ingressgateway_xfcc_header_validate_Status	Https Status value after performing xfccHeaderValidation at Ingress Gateway.	200 (OK), 400 (BAD_REQUEST)
oc_ingressgateway_xfcc_header_validate_Cause	This tag determines the validation cause for the xfcc header validation metric being pegged.	VALIDATION_FAILURE, VALIDATION_SUCCESS, HEADER_NOT_FOUND
oc_ingressgateway_xfcc_header_validate_CertsCompared	This tag captures the total number of certificates compared in XFCC header at ingress gateway during the header validation.	Count of the certificates compared (0,1,2..)
oc_configclient_request_total_releaseVersion	This tag indicates the current release version of ingress gateway.	Picked from helm chart{{ .Chart.Version }}
oc_configclient_request_total_configVersion	This tag indicates the configuration version that ingress gateway is currently maintaining.	Initial value is 0. Incremental value received from config server whenever there is an update from config server (0, 1, 2...)
oc_configclient_response_total_releaseVersion	This tag indicates the current release version of ingress gateway.	Picked from helm chart {{ .Chart.Version }}
oc_configclient_response_total_configVersion	This tag indicates the configuration version that ingress gateway is currently maintaining.	Value received from config server (1, 2...)

Tags	Description	Possible Values
oc_configclient_response_total_updated	This tag indicates whether the configuration was updated or not.	true/false

Ingress Gateway Metrics

Table 6-110 oc_ingressgateway_http_requests_total

Field	Details
Description	This metric is pegged as soon as the request reaches the Ingress gateway in the first custom filter of the application.
Type	Counter
Dimension	<ul style="list-style-type: none"> • NFType • NFServiceType • Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • ClientCertIdentity

Table 6-111 oc_ingressgateway_http_responses_total

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Status • Method • Route_path • NFType • NFServiceType • Host • HttpVersion • Scheme • Identifier • ClientCertIdentity

Table 6-112 oc_ingressgateway_request_latency_seconds

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF. This metric tracks the amount of time taken for processing the request. It starts as soon the request reaches the first custom filter of the application and lasts till the response is sent back to the consumer NF from the last custom filter of the application.
Type	Timer
Dimension	<ul style="list-style-type: none"> • quantile • InstanceIdentifier

Table 6-113 oc_ingressgateway_connection_failure_total

Field	Details
Description	This metric is pegged in the customized Jetty Client as soon as it fails to connect to the destination service with direction as ingressOut. Here in case of Ingress gateway, the destination service is a backend microservice of the NF. And TLS connection failure metrics when connecting to ingress with direction as ingress.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Host • Port • Direction • InstanceIdentifier • error_reason

Table 6-114 oc_ingressgateway_global_ratelimit_total

Field	Details
Description	This metric is pegged in the custom filter implemented to check the global rate limit conditions.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Route_path • Scheme • InstanceIdentifier • Status (Rate limit Status field is different here)

Table 6-115 oc_ingressgateway_route_ratelimit_total

Field	Details
Description	This metric is pegged in the custom filter implemented to check the route level rate limit conditions.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Method • Route_path • Scheme • InstanceIdentifier • Status (Rate limit Status field is different here)

Table 6-116 oc_ingressgateway_request_processing_latency_seconds

Field	Details
Description	This metric is pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF. This metric captures the amount of time taken for processing of the request only within Ingress gateway. It starts as soon the request reaches the first custom filter of the application and lasts till the request is forwarded to the destination.
Type	Timer

Table 6-116 (Cont.) oc_ingressgateway_request_processing_latency_seconds

Field	Details
Dimension	<ul style="list-style-type: none"> quantile InstanceIdentifier

Table 6-117 oc_ingressgateway_jetty_request_stat_metrics_total

Field	Details
Description	This metric is pegged for every event occurred when a request is sent to IGW.
Type	Counter
Dimension	<ul style="list-style-type: none"> event client_type InstanceIdentifier

Table 6-118 oc_ingressgateway_jetty_response_stat_metrics_total

Field	Details
Description	This metric is pegged for every event occurred when a response is received by IGW
Type	Counter
Dimension	<ul style="list-style-type: none"> event client_type InstanceIdentifier

Table 6-119 oc_ingressgateway_jetty_latency_seconds

Field	Details
Description	This metric is pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client.
Type	Timer
Dimension	<ul style="list-style-type: none"> quantile InstanceIdentifier

Table 6-120 oc_ingressgateway_netty_latency_seconds

Field	Details
Description	This metric is pegged in Netty outbound handler that captures the amount of time taken for processing of the request by netty server.
Type	Timer
Dimension	<ul style="list-style-type: none"> quantile InstanceIdentifier

Table 6-121 oc_ingressgateway_request_content_metrics_total

Field	Details
Description	This metric is pegged by default filter RequestContentMetrics. It pegs whether request has request body or not.
Type	Counter
Dimension	<ul style="list-style-type: none"> • method • content_available • InstanceIdentifier

Table 6-122 oc_ingressgateway_xfcc_header_validate_total

Field	Details
Description	This metric is pegged when xfccHeaderValidation is enabled in XfccHeaderValidationFilter. This metric along with the specified dimension captures the successful/ un-successful validation of XFCC header in the incoming request.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Route_path • Status • Cause • CertsCompared • InstanceIdentifier • ErrorOriginator

Table 6-123 oc_configclient_request_total

Field	Details
Description	This metric is pegged whenever config client is polling for configuration update from common configuration server.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Release version • Config version

Table 6-124 oc_configclient_response_total

Field	Details
Description	This metrics is pegged whenever config client receives response from common configuration server.
Type	Counter
Dimension	<ul style="list-style-type: none"> • Release version • Config version • Updated

OAuth Metrics

OAuth Metrics Common Tags

Tags	Description	Possible Values
scope	NF service name(s) of the NF service producer(s), separated by white spaces.	NA
issuer	NF instance id of NRF.	NA
subject	NF instance id of service consumer.	NA
reason	reason contains the human readable message for oauth validation failure.	NA

Below are the metrics and their respective tags that are available in Oauth:

Table 6-125 oc_oauth_validation_successful_total

Field	Details
Description	This metric is pegged in the OAuth validator implementation if the received OAuth token is validated successfully. The implementation of OAuth validator is used in Ingress Gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> • issuer • subject • scope

Table 6-126 oc_oauth_validation_failure_total

Field	Details
Description	This metric is pegged in the implementation of OAuth validator if the validation of the the received OAuth token fails. The implementation of OAuth validator is used in Ingress Gateway.
Type	Counter
Dimension	<ul style="list-style-type: none"> • issuer • subject • scope • reason

6.6 Egress Gateway Metrics

Egress Gateway Metrics

The following table describes the Egress Gateway Metrics.

Table 6-127 oc_egressgateway_http_requests_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • Method • NFType • NFServiceType • Host • HttpVersion • Scheme • Proxy • InstanceIdentifier
Pegging Instance	This metric is pegged as soon as the request reaches the Egress Gateway in the first custom filter of the application.

Table 6-128 oc_egressgateway_http_responses_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • Status • Method • NFType • NFServiceType • Host • HttpVersion • Scheme • InstanceIdentifier • Direction • BlacklistedFqdn
Pegging Instance	<p>This metric will be pegged in the last custom filter of the Egress gateway while the response is being sent back to backend NF microservice with direction as egress.</p> <p>This will also be pegged when the response is fetched in Jetty responseListener with direction as egressOut.</p> <p>BlacklistedFqdn tag will be filled with BlacklistedFqdn when request is sent with blacklisted producer.</p>

Table 6-129 oc_egressgateway_request_latency_seconds

Field	Details
Available Tags	<ul style="list-style-type: none"> • quantile • InstanceIdentifier
Pegging Instance	This metric is pegged in the last custom filter of the Ingress Gateway while the response is being sent back to the consumer NF. This metric tracks the amount of time taken for processing the request. It starts as soon as the request reaches the first custom filter of the application and lasts till, the response is sent back to the the consumer NF from the last custom filter of the application.

Table 6-130 oc_egressgateway_connection_failure_total

Field	Details
Available Tags	<ul style="list-style-type: none"> Host Port InstanceIdentifier Direction error_reason
Pegging Instance	<p>This metric will be pegged in the customized Jetty Client as soon as it fails to connect to the destination service. Here in case of Egress gateway, the destination service will be Producer NF.</p> <p>This will also be pegged when the request to Producer NF fails in Jetty request Listener with direction as egressOut</p>

Table 6-131 oc_egressgateway_notification_ratelimit_total

Field	Details
Available Tags	<ul style="list-style-type: none"> Method Scheme InstanceIdentifier
Pegging Instance	<p>This metric is pegged in the custom filter implemented to check the notification rate limit conditions.</p>

Table 6-132 oc_egressgateway_request_processing_latency_seconds

Field	Details
Available Tags	<ul style="list-style-type: none"> quantile InstanceIdentifier
Pegging Instance	<p>This metric is pegged in the last custom filter of the Egress Gateway while the response is sent back to the consumer NF. This metric tracks the amount of time taken for processing the request only within Egress Gateway. It starts as soon as the request reaches the first custom filter of the application and lasts till the request is forwarded to the destination.</p>

Table 6-133 oc_egressgateway_jetty_request_stat_metrics_total

Field	Details
Available Tags	<ul style="list-style-type: none"> event client_type InstanceIdentifier
Pegging Instance	<p>This metric is pegged for every event occurred when a request is sent to EGW</p>

Table 6-134 oc_egressgateway_jetty_response_stat_metrics_total

Field	Details
Available Tags	<ul style="list-style-type: none"> event client_type InstanceIdentifier
Pegging Instance	This metric is pegged for every event occurred when a response is received by EGW

Table 6-135 oc_egressgateway_jetty_response_stat_metrics_total

Field	Details
Available Tags	<ul style="list-style-type: none"> event client_type InstanceIdentifier
Pegging Instance	This metric is pegged for every event occurred when a response is received by EGW

Table 6-136 oc_egressgateway_jetty_latency_seconds

Field	Details
Available Tags	<ul style="list-style-type: none"> quantile InstanceIdentifier
Pegging Instance	This metric is pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client

Table 6-137 oc_egressgateway_jetty_latency_seconds

Field	Details
Available Tags	<ul style="list-style-type: none"> quantile InstanceIdentifier
Pegging Instance	This metric is pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client

Table 6-138 oc_egressgateway_netty_latency_seconds

Field	Details
Available Tags	<ul style="list-style-type: none"> quantile InstanceIdentifier
Pegging Instance	This metric is pegged in Netty outbound handler that captures the amount of time taken for processing of the request by netty server

Table 6-139 oc_egressgateway_request_content_metrics_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • method • content_available • InstanceIdentifier
Pegging Instance	This metric is pegged by default filter RequestContentMetrics. It pegs whether request has request body or not and the method.

Table 6-140 oc_egressgateway_blacklisted_producer_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • NFType • NFServiceType • InstanceIdentifier • Host • Route_path
Pegging Instance	This metric is a counter. Track number of times producer is blacklisted.

Table 6-141 oc_configclient_request_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • Release version • Config version
Pegging Instance	This metric will be pegged whenever config client is polling for configuration update from common configuration server

Table 6-142 oc_configclient_response_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • Release version • Config version • Updated
Pegging Instance	This metrics will be pegged whenever config client receives response from common configuration server

Egress Gateway Metrics Common Tags

The following table describes the common tags used in Egress Gateway Metrics.

Table 6-143 Method

Field	Details
Available Tags	Http method
Pegging Instance	GET, PUT, POST, DELETE, PATCH

Table 6-144 NFType

Field	Details
Available Tags	Name of the NF Type
Pegging Instance	"UNKNOWN" (Updates are available when Ingress is 5G aware)

Table 6-145 NFServiceType

Field	Details
Available Tags	Name of the Service within the NF
Pegging Instance	"UNKNOWN" (Updates are available when Ingress is 5G aware)

Table 6-146 Host

Field	Details
Available Tags	(IP or fqdn): port of ingress gateway
Pegging Instance	Not Applicable

Table 6-147 HttpVersion

Field	Details
Available Tags	Http protocol version (http1.1/ http2)
Pegging Instance	HTTP1.1, HTTP2.0

Table 6-148 Scheme

Field	Details
Available Tags	Http protocol scheme (http/https)
Pegging Instance	HTTP, HTTPS, UNKNOWN

Table 6-149 Proxy

Field	Details
Available Tags	Value received for "x-custom-egress-proxy-header".
Pegging Instance	Unknown or value of "x-custom-egress-proxy-header".

Table 6-150 oc_egressgateway_connection_failure_Host

Field	Details
Available Tags	destination ip/fqdn
Pegging Instance	Not Applicable

Table 6-151 oc_egressgateway_connection_failure_Port

Field	Details
Available Tags	destination port
Pegging Instance	Not Applicable

Table 6-152 BlacklistedFqdn

Field	Details
Available Tags	Blacklisted Producer Fqdn
Pegging Instance	Unknown or Blacklisted Producer Fqdn

Table 6-153 oc_configclient_request_total_releaseVersion

Field	Details
Available Tags	This tag indicates the current release version of egress gateway
Pegging Instance	Picked from helm chart{{ .Chart.Version }}

Table 6-154 oc_configclient_request_total_configVersion

Field	Details
Available Tags	This tag indicates the configuration version that egress gateway is currently maintaining
Pegging Instance	Initial value is 0. Incremental value received from config server whenever there is an update from config server (0, 1, 2...)

Table 6-155 oc_configclient_response_total_releaseVersion

Field	Details
Available Tags	This tag indicates the current release version of egress gateway
Pegging Instance	Picked from helm chart{{ .Chart.Version }}

Table 6-156 oc_configclient_response_total_configVersion

Field	Details
Available Tags	This tag indicates the configuration version that egress gateway is currently maintaining
Pegging Instance	Value received from config server (1, 2...)

Table 6-157 oc_configclient_response_total_updated

Field	Details
Available Tags	This tag indicates whether the configuration was updated or not

Table 6-157 (Cont.) oc_configclient_response_total_updated

Field	Details
Pegging Instance	true/false

SCP Metrics

The following table describes the different metrics and their respective tags that are available in the SCP Module:

Table 6-158 oc_egressgateway_scp_http_requests_total

Field	Details
Available Tags	<ul style="list-style-type: none"> Scp_Fqdn Reroute_Path Response_Code (This would be populated as blank for requests) Attempt HttpVersion Scheme InstanceIdentifier
Pegging Instance	This metric is pegged in the ScpFilter only when SCP Integration is enabled.

Table 6-159 oc_egressgateway_scp_http_responses_total

Field	Details
Available Tags	<ul style="list-style-type: none"> Scp_Fqdn Reroute_Path Response_Code Attempt HttpVersion Scheme InstanceIdentifier
Pegging Instance	This metric is pegged in the ScpFilter only when Scp Integration is enabled. It is also being pegged in the Scp Retry Filter when Scp re-route feature is enabled.

SCP Metrics common tags

The following table describes common tags used in SCP Metrics.

Table 6-160 Scp_Fqdn

Field	Details
Description	SCP Fqdn
Possible Values	Not Applicable

Table 6-161 Reroute_Path

Field	Details
Description	Path that matched the request to over corresponding route Example: /nef/**
Possible Values	Not Applicable

Table 6-162 Response_Code

Field	Details
Description	It is populated as blank for request metrics. During failure scenario's, it is populated with "ERROR" for response metrics. Example: ERROR, OK
Possible Values	ERROR, NOT ACCEPTABLE, OK

Table 6-163 Attempt

Field	Details
Description	Attempt number for scp re-route. Example: 1, 2 etc.
Possible Values	Not Applicable

Table 6-164 HttpVersion

Field	Details
Description	Http protocol version (http1.1/ http2)
Possible Values	HTTP/1.1, HTTP/2.0

Table 6-165 Scheme

Field	Details
Description	Http protocol scheme (http/https)
Possible Values	HTTP, HTTPS, UNKNOWN

Table 6-166 InstanceIdentifier

Field	Details
Description	Prefix of the pod configured in helm when there are multiple instances in same deployment
Possible Values	Prefix configured in helm otherwise UNKNOWN

Oauth Metrics

The following table includes metrics and their respective tags that are available in the Oauth Module:

Table 6-167 oc_oauth_nrf_request_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • NrfFqdn
Pegging Instance	This metric is pegged in the OAuth client implementation if the request is sent to NRF for requesting the OAuth token. OAuth client implementation will be used in Egress gateway.

Table 6-168 oc_oauth_nrf_response_success_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • StatusCode • NrfFqdn
Pegging Instance	This metric is pegged in the OAuth client implementation if an OAuth token is successfully received from the NRF. OAuth client implementation is used in the Egress Gateway.

Table 6-169 oc_oauth_nrf_response_failure_total

Field	Details
Available Tags	<ul style="list-style-type: none"> • ConsumerNFInstanceId • ConsumerNFType • TargetNFType • TargetNFInstanceId • scope • StatusCode • ErrorOriginator • NrfFqdn
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever GetAccessTokenFailedException is caught.

Table 6-170 oc_oauth_request_failed_internal_total

Field	Details
Available Tags	<ul style="list-style-type: none"> ConsumerNFInstanceId ConsumerNFType TargetNFType TargetNFInstanceId scope StatusCode ErrorOriginator NrfFqdn
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever InternalServerErrorException is caught.

Table 6-171 oc_oauth_token_cache_total

Field	Details
Available Tags	<ul style="list-style-type: none"> ConsumerNFInstanceId ConsumerNFType TargetNFType TargetNFInstanceId scope
Pegging Instance	This metric is pegged in the OAuth Client Implementation if the OAuth token is found in the cache.

Table 6-172 oc_oauth_request_invalid_total

Field	Details
Available Tags	<ul style="list-style-type: none"> ConsumerNFInstanceId ConsumerNFType TargetNFType TargetNFInstanceId scope StatusCode ErrorOriginator
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever a BadAccessTokenRequestException/JsonProcessingException is caught.

Table 6-173 oc_egressgateway_oauth_access_token_request_header_missing

Field	Details
Available Tags	NA
Pegging Instance	This metric is pegged in the OAuthClientFilter in Egress Gateway whenever oc-access-token-request-info header is missing in the request.

Table 6-174 oc_oauth_cert_expiryStatus

Field	Details
Available Tags	<ul style="list-style-type: none"> • id • certificateName • secretName
Pegging Instance	This Gauge metric is used to peg expiry date of the certificate. This metric is further used for raising alarms if certificate expires within 30 days or 7 days.

Table 6-175 oc_oauth_cert_loadStatus

Field	Details
Available Tags	<ul style="list-style-type: none"> • id • certificateName • secretName
Pegging Instance	This gauge metric is used to peg expiry date of the certificate. This metric is further used for raising alarms if certificate expires within 30 days or 7 days.

Table 6-176 oc_oauth_request_failed_cert_expiry

Field	Details
Available Tags	<ul style="list-style-type: none"> • target nf type • target nf instance id • consumer nf instance id • nrf instance id • service name of nf producer service • key id
Pegging Instance	This counter metric is used to keep track of number of requests with keyId in token that failed due to certificate expiry. It is pegged whenever OAuth Validator module throws OAuth custom exception due to certificate expiry for an incoming request.

Table 6-177 oc_oauth_keyid_count

Field	Details
Available Tags	<ul style="list-style-type: none"> • target nf type • target nf instance id • consumer nf instance id • nrf instance id • service name of nf producer service • key id
Pegging Instance	This counter metric used to keep track of number of requests received with keyId in token. It is pegged whenever a request with an access token containing keyid in header comes to OAuth Validator.

Table 6-178 oc_oauth_nrf_token_retrieval_failure_total

Field	Details
Available Tags	<ul style="list-style-type: none"> ConsumerNFInstanceId ConsumerNFType TargetNFType TargetNFInstanceId scope StatusCode ErrorOriginator ErrorDetail NrfFqdn
Pegging Instance	This metric is pegged to track requests discarded due to oAuth token retrieval failure from NRF.

OAuth Metrics (NRF-Client Mgmt Service Call-Flow)**Table 6-179 oc_oauth_nrf_client_subscription_request_total**

Field	Details
Available Tags	<ul style="list-style-type: none"> NrfClientUrl EgwNotificationUrl
When it is pegged	<p>This metric will be pegged in the OAuth client implementation module</p> <p>when a subscription request is sent from EGW to NRF-Client Mgmt Svc</p> <p>with request-URL (NrfClientUrl) and request body containing notification-URL of EGW (EgwNotificationUrl) for OAuth Client notification requests generated from NRF-Client Mgmt Svc to EGW. OAuth client implementation will be used in Egress gateway.</p>

Table 6-180 oc_oauth_nrf_client_notification_request_total

Field	Details
Available Tags	None
When it is pegged	<p>This metric will be pegged in the OAuth client implementation module when a notification request is sent from NRF-Client Mgmt Svc to EGW. OAuth client implementation will be used in Egress gateway.</p>

Table 6-181 oc_oauth_nrf_client_subscription_response_total

Field	Details
Available Tags	<ul style="list-style-type: none"> NrfClientUrl EgwNotificationUrl StatusCode

Table 6-181 (Cont.) oc_oauth_nrf_client_subscription_response_total

Field	Details
When it is pegged	This metric will be pegged in the OAuth client implementation module when a subscription response is sent from NRF-Client Mgmt Svc having URL (NrfClientUrl) to EGW having URL (EgwNotificationUrl). StatusCode tag will capture the corresponding response status obtained from NRF-Client. OAuth client implementation will be used in Egress gateway.

Table 6-182 oc_oauth_nrf_client_notification_response_total

Field	Details
Available Tags	<ul style="list-style-type: none"> StatusCode
When it is pegged	This metric will be pegged in the OAuth client implementation module when a notification response is sent from EGW to NRF-Client Mgmt Svc. StatusCode tag will capture the corresponding response status sent from EGW. OAuth client implementation will be used in Egress gateway.

Table 6-183 oc_oauth_nrf_client_active_nrf_instances

Field	Details
Available Tags	<ul style="list-style-type: none"> NrfFqdn
When it is pegged	This is a GAUGE metric which keeps track of healthy NRF Fqdns received from NRF-Client Mgmt Svc as part of subscription response/ notification request to EGW.

OAuth Metrics common tags

The following table describes common tags used in the OAuth Module.

Table 6-184 ConsumerNFInstanceId

Field	Details
Description	NF instance id of the NF service consumer
Possible Values	Not Applicable

Table 6-185 ConsumerNFType

Field	Details
Description	The NF type of the NF service consumer

Table 6-185 (Cont.) ConsumerNFType

Field	Details
Possible Values	NRF, UDM, AMF, SMF, AUSF, NEF, PCF, SMSF, NSSF, UDR, LMF, GMLC, 5G_EIR, SEPP, UPF, N3IWF, AF, UDSF, BSF, CHF, NWDAF

Table 6-186 TargetNFType

Field	Details
Description	The NF type of the NF service producer
Possible Values	NRF, UDM, AMF, SMF, AUSF, NEF, PCF, SMSF, NSSF, UDR, LMF, GMLC, 5G_EIR, SEPP, UPF, N3IWF, AF, UDSF, BSF, CHF, NWDAF

Table 6-187 TargetNFInstanceId

Field	Details
Description	NF instance id of the NF service producer
Possible Values	Not Applicable

Table 6-188 scope

Field	Details
Description	NF service name(s) of the NF service producer(s), separated by whitespaces
Possible Values	Not Applicable

Table 6-189 StatusCode

Field	Details
Description	Status code of NRF access token request
Possible Values	Bad Request, Internal Server Error etc. (HttpStatus.*)

Table 6-190 ErrorOriginator

Field	Details
Description	from where error is originated (nrf or egress)
Possible Values	Nrf, EgressGW

Table 6-191 issuer

Field	Details
Description	NF instance id of NRF
Possible Values	Not Applicable

Table 6-192 subject

Field	Details
Description	NF instance id of service consumer
Possible Values	Not Applicable

Table 6-193 reason

Field	Details
Description	reason contains the human readable msg for oauth validation failure
Possible Values	Not Applicable

Table 6-194 NrfFqdn

Field	Details
Description	NrfFqdn tag determines the corresponding fqdn of NRF where the request has been forwarded to.
Possible Values	Nrf-Fqdn (dynamic value based on Fqdn), NA

Table 6-195 NrfClientUrl

Field	Details
Description	This tag determines the url of NRF-Client Mgmt Svc where subscription requests are sent from OAuth Client module in EGW.
Possible Values	URL of NRF-Client Mgmt Svc (Dynamic value)

Table 6-196 EgwNotificationUrl

Field	Details
Description	This tag determines the notification URL mapped in OAuth Client module of EGW where NRF-Client Mgmt Svc will send notifications requests.
Possible Values	Notification URL (Dynamic value)

Table 6-197 ConfigurationType

Field	Details
Description	This tag determines the type of configuration in place for OAuth Client in Egress Gateway. If nrfClientQueryEnabled Helm parameter in oauthClient Helm configurations at Egress Gateway is false then the ConfigurationType is STATIC, else DYNAMIC.
Possible Values	STATIC, DYNAMIC

Table 6-198 oc_egressgateway_msgcopy_requests_total

Field	Details
Description	This is incremented whenever egress request message is sent or acknowledged from Kafka.
Type	Counter
Dimension	
Example	

Table 6-199 oc_egressgateway_msgcopy_responses_total

Field	Details
Description	This is incremented whenever egress response message is sent or acknowledged from Kafka.
Type	Counter
Dimension	
Example	

7

Alerts

This section provides information on Oracle Communications Network Exposure Function (NEF) alerts and their configuration. The section contains the following sections:

- [Configuring Alerts](#)
- [List of Alerts](#)

Note

The performance and capacity of the NEF system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.

7.1 List of Alerts

This section provides detailed information about the alert rules defined for NEF. It consists of the following two types of alerts

1. NEF Alerts - Contains the system level and application level alerts of NEF.
2. CAPIF Alerts - Contains the system level and application level alerts of CAPIF.

7.1.1 NEF Alerts

This chapter includes information about the following NEF alerts:

- [System Level Alerts](#)
- [Application Level Alerts](#)

Note

- The performance and capacity of the NEF system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.
- Due to unavailability of metric and/or MQL queries, the following alerts are not supported for OCI:
 - OcnfNfStatusUnavailable
 - OcnfPodsRestart
 - OcnfIngressGatewayServiceDown
 - OcnfApiRouterServiceDown
 - OcnfFiveGcAgentServiceDown
 - OcnfMonitoringEventServiceDown
 - OcnfCCFClientServiceDown
 - OcnfExpiryAuditorServiceDown
 - OcnfQOSServiceDown
 - OcnfTIServiceDown
 - OcnfDTServiceDown
 - OcnfEgressGatewayServiceDown
 - OcnfMemoryUsageCrossedMinorThreshold
 - OcnfMemoryUsageCrossedMajorThreshold
 - FiveGcInvalidConfiguration
 - OcnfAllSiteStatus
 - OcnfDBReplicationStatus

7.1.1.1 System Level Alerts

This section lists the system level alerts for NEF.

7.1.1.1.1 OcnfNfStatusUnavailable

Table 7-1 OcnfNfStatusUnavailable

Field	Details
Description	'NEF services unavailable'
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{ . first value humanizeTimestamp }}{{ end }} : All NEF services are unavailable."
Severity	Critical
Condition	All the NEF services are unavailable, either because the NEF is getting deployed or purged.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7001

Table 7-1 (Cont.) OcnfNfStatusUnavailable

Field	Details
Metric Used	<p>'up'</p> <p>Note: This is a Prometheus metric used for instance availability monitoring.</p> <p>If this metric is not available, use a similar metric as exposed by the monitoring system.</p>
Recommended Actions	<p>The alert is cleared automatically when the NEF services restart.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for service-specific alerts which may be causing the issues with service exposure. 2. Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> <ol style="list-style-type: none"> a. Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> 3. Refer to the application logs on Kibana and check for database related failures such as connectivity and invalid secrets. The logs can be filtered based on the services. 4. Check for helm status to make sure there are no errors: <pre>\$ helm status <helm release name of the desired NF> -n <namespace></pre> <p>If it is not in "STATUS : DEPLOYED", then capture logs and event again.</p> 5. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support. <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on the Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.1.1.2 OcnfPodsRestart

Table 7-2 OcnfPodsRestart

Field	Details
Description	'Pod <Pod Name> has restarted.
Summary	"kubernetes_namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : A Pod has restarted"

Table 7-2 (Cont.) OcnefPodsRestart

Field	Details
Severity	Major
Condition	A pod belonging to any of the NEF services has restarted.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7002
Metric Used	kube_pod_container_status_restarts_total
Recommended Actions	<p>The alert is cleared automatically if the specific pod is up.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer to the application logs on Kibana and filter based on pod name, check for database related failures such as connectivity and Kubernetes secrets. 2. To check the orchestration logs for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> a. Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> b. Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support. <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on the Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.1.1.3 OcnefTotalExternalIngressTrafficRateAboveMinorThreshold

Table 7-3 OcnefTotalExternalIngressTrafficRateAboveMinorThreshold

Field	Details
Description	OCNEF External Ingress traffic rate is above the configured minor threshold i.e. 800 TPS (current value is: {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 80 percent of max TPS (1000)"
Severity	Minor
Condition	<p>The total NEF External Ingress traffic rate has crossed the configured minor threshold of 800 TPS.</p> <p>Default value of this alert trigger point in NefAlertrules alert file is 80 % of 1000 (maximum ingress request rate).</p>

Table 7-3 (Cont.) OcnefTotalExternalIngressTrafficRateAboveMinorThreshold

Field	Details
OID	1.3.6.1.4.1.323.5.3.39.1.2.7003
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the total External Ingress traffic rate falls below the minor threshold or when the total traffic rate crosses the major threshold, in which case the OcnefTotalExternalIngressTrafficRateAboveMajorThreshold alert is raised.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file. Reassess why the NEF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.1.1.4 OcnefTotalFivegcIngressTrafficRateAboveMinorThreshold

Table 7-4 OcnefTotalFivegcIngressTrafficRateAboveMinorThreshold

Field	Details
Description	OCNEF Fivegc Ingress traffic rate is above the configured minor threshold i.e. 800 TPS (current value is: {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 80 percent of max TPS (1000)"
Severity	Minor
Condition	<p>The total NEF Fivegc Ingress traffic rate has crossed the configured minor threshold of 800 TPS.</p> <p>Default value of this alert trigger point in NefAlertrules alert file is 80 % of 1000 (maximum ingress request rate).</p>
OID	1.3.6.1.4.1.323.5.3.39.1.2.7004
Metric Used	oc_ingressgateway_http_requests_total

Table 7-4 (Cont.) OcnfTotalFivegcIngressTrafficRateAboveMinorThreshold

Field	Details
Recommended Actions	<p>The alert is cleared either when the total Fivegc Ingress traffic rate falls below the minor threshold or when the total traffic rate crosses the major threshold, in which case the OcnfTotalFivegcIngressTrafficRateAboveMajorThreshold alert is raised.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>Reassess why the NEF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.1.1.5 OcnfTotalExternalIngressTrafficRateAboveMajorThreshold

Table 7-5 OcnfTotalExternalIngressTrafficRateAboveMajorThreshold

Field	Details
Description	OCNEF External Ingress traffic rate is above the configured major threshold i.e. 900 TPS (current value is: {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 90 percent of max TPS (1000)"
Severity	Major
Condition	<p>The total NEF External Ingress traffic rate has crossed the configured major threshold of 900 TPS.</p> <p>Default value of this alert trigger point in NefAlertrules alert file is 90 % of 1000 (maximum ingress request rate).</p>
OID	1.3.6.1.4.1.323.5.3.39.1.2.7005
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the total External Ingress traffic rate falls below the major threshold or when the total traffic rate crosses the critical threshold, in which case the OcnfTotalExternalIngressTrafficRateAboveCriticalThreshold alert is raised.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>Reassess why the NEF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.1.1.6 OcnefTotalFivegcIngressTrafficRateAboveMajorThreshold

Table 7-6 OcnefTotalFivegcIngressTrafficRateAboveMajorThreshold

Field	Details
Description	OCNEF Fivegc Ingress traffic rate is above the configured major threshold i.e. 900 TPS (current value is: {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 90 percent of max TPS (1000)"
Severity	Major
Condition	The total NEF Fivegc Ingress traffic rate has crossed the configured major threshold of 900 TPS. Default value of this alert trigger point in NefAlertrules alert file is 90 % of 1000 (maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7006
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the total Fivegc Ingress traffic rate falls below the major threshold or when the total traffic rate crosses the critical threshold, in which case the OcnefTotalFivegcIngressTrafficRateAboveCriticalThreshold alert is raised.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>Reassess why the NEF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.1.1.7 OcnefTotalExternalIngressTrafficRateAboveCriticalThreshold

Table 7-7 OcnefTotalExternalIngressTrafficRateAboveCriticalThreshold

Field	Details
Description	OCNEF External Ingress traffic rate is above the configured critical threshold i.e. 950 TPS (current value is: {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 95 percent of max TPS (1000)"
Severity	Critical
Condition	The total NEF External Ingress traffic rate has crossed the configured critical threshold of 950 TPS. Default value of this alert trigger point in NefAlertrules alert file is 95 % of 1000 (maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7007
Metric Used	oc_ingressgateway_http_requests_total

Table 7-7 (Cont.) OcnefTotalExternalIngressTrafficRateAboveCriticalThreshold

Field	Details
Recommended Actions	<p>The alert is cleared either when the total External Ingress traffic rate falls below the critical threshold.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>Reassess why the NEF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.1.1.8 OcnefTotalFivegcIngressTrafficRateAboveCriticalThreshold

Table 7-8 OcnefTotalFivegcIngressTrafficRateAboveCriticalThreshold

Field	Details
Description	OCNEF Fivegc Ingress traffic rate is above the configured critical threshold i.e. 950 TPS (current value is: {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 95 percent of max TPS (1000)"
Severity	Critical
Condition	<p>The total NEF Fivegc Ingress traffic rate has crossed the configured critical threshold of 950 TPS.</p> <p>Default value of this alert trigger point in NefAlertrules alert file is 95 % of 1000 (maximum ingress request rate).</p>
OID	1.3.6.1.4.1.323.5.3.39.1.2.7008
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the total Fivegc Ingress traffic rate falls below the critical threshold.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>Reassess why the NEF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.1.1.9 OcnefExternalIngressTransactionErrorRateAboveZeroPointOnePercent

Table 7-9 OcnefExternalIngressTransactionErrorRateAboveZeroPointOnePercent

Field	Details
Description	External Ingress transaction Error rate is above 0.1 percent(current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction Error rate detected above 0.1 percent of total transactions"
Severity	Warning
Condition	The number of failed external ingress transactions is above 0.1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7009
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure external ingress 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 OcnefExternalIngressTransactionErrorRateAbove1Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.10 OcnefFivegcIngressTransactionErrorRateAboveZeroPointOnePercent

Table 7-10 OcnefFivegcIngressTransactionErrorRateAboveZeroPointOnePercent

Field	Details
Description	Fivegc Ingress transaction error rate is above 0.1 percent of total transactions (current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 0.1 percent of total transactions"
Severity	Warning
Condition	The number of failed Fivegc ingress transactions is above 0.1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7010
Metric Used	oc_ingressgateway_http_responses_total

Table 7-10 (Cont.)
OcnefFivegcIngressTransactionErrorRateAboveZeroPointOnePercent

Field	Details
Recommended Actions	<p>The alert is cleared when the number of failure Fivegc ingress 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 OcnefFivegcIngressTransactionErrorRateAbove1Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.11 OcnefExternalIngressTransactionErrorRateAbove1Percent

Table 7-11 OcnefExternalIngressTransactionErrorRateAbove1Percent

Field	Details
Description	External Ingress transaction error rate is above 1 percent of total transactions (current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 1 percent of total transactions"
Severity	Warning
Condition	The number of failed External Ingress transactions is above 1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7011
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 1 percent of the total transactions or when the number of failed transactions crosses the 10% threshold, in which case the OcnefExternalIngressTransactionErrorRateAbove10Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.12 OcnefFivegcIngressTransactionErrorRateAbove1Percent

Table 7-12 OcnefFivegcIngressTransactionErrorRateAbove1Percent

Field	Details
Description	Fivegc Ingress transaction error rate is above 1 percent of total Fivegc Ingress transactions (current value is {{ \$value }})

Table 7-12 (Cont.) OcnefFivegcIngressTransactionErrorRateAbove1Percent

Field	Details
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 1 percent of total transactions"
Severity	Warning
Condition	The number of failed Fivegc Ingress transactions is above 1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7012
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Fivegc Ingress transactions is below 1 percent of the total transactions or when the number of failed transactions crosses the 10% threshold, in which case the OcnefFivegcIngressTransactionErrorRateAbove10Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.13 OcnefExternalIngressTransactionErrorRateAbove10Percent

Table 7-13 OcnefExternalIngressTransactionErrorRateAbove10Percent

Field	Details
Description	External Ingress transaction error rate is above 10 percent of total External Ingress transactions (current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 10 percent of total transactions"
Severity	Minor
Condition	The number of failed External Ingress transactions is above 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7013
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 10 percent of the total transactions or when the number of failed transactions crosses the 25% threshold, in which case the OcnefExternalIngressTransactionErrorRateAbove25Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.14 OcnefFivegcIngressTransactionErrorRateAbove10Percent

Table 7-14 OcnefFivegcIngressTransactionErrorRateAbove10Percent

Field	Details
Description	Fivegc Ingress transaction error rate is above 10 percent of total Fivegc Ingress transactions (current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 10 percent of total transactions"
Severity	Minor
Condition	The number of failed Fivegc Ingress transactions is above 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7014
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Fivegc Ingress transactions is below 10 percent of the total transactions or when the number of failed transactions crosses the 25% threshold, in which case the OcnefFivegcIngressTransactionErrorRateAbove25Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.15 OcnefExternalIngressTransactionErrorRateAbove25Percent

Table 7-15 OcnefExternalIngressTransactionErrorRateAbove25Percent

Field	Details
Description	External Ingress transaction error rate detected above 25 percent of total External Ingress transactions (current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 25 percent of total transactions"
Severity	Major
Condition	The number of failed External Ingress transactions is above 25 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7015
Metric Used	oc_ingressgateway_http_responses_total

Table 7-15 (Cont.) OcnefExternalIngressTransactionErrorRateAbove25Percent

Field	Details
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 25 percent of the total transactions or when the number of failed transactions crosses the 50% threshold, in which case the OcnefExternalIngressTransactionErrorRateAbove50Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.16 OcnefFivegcIngressTransactionErrorRateAbove25Percent

Table 7-16 OcnefFivegcIngressTransactionErrorRateAbove25Percent

Field	Details
Description	Fivegc Ingress transaction error rate detected above 25 percent of total Fivegc Ingress transactions (current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 25 percent of total transactions"
Severity	Major
Condition	The number of failed Fivegc Ingress transactions is above 25 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7016
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Fivegc Ingress transactions is below 25 percent of the total transactions or when the number of failed transactions crosses the 50% threshold, in which case the OcnefFivegcIngressTransactionErrorRateAbove50Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.17 OcnefExternalIngressTransactionErrorRateAbove50Percent

Table 7-17 OcnefExternalIngressTransactionErrorRateAbove50Percent

Field	Details
Description	External Ingress transaction error rate detected above 50 percent of total External Ingress transactions (current value is {{ \$value }})

Table 7-17 (Cont.) OcnefExternalIngressTransactionErrorRateAbove50Percent

Field	Details
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 50 percent of total transactions"
Severity	Critical
Condition	The number of failed External Ingress transactions is above 50 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7017
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 50 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.18 OcnefFivegcIngressTransactionErrorRateAbove50Percent

Table 7-18 OcnefFivegcIngressTransactionErrorRateAbove50Percent

Field	Details
Description	Fivegc Ingress transaction error rate detected above 50 percent of total Fivegc Ingress transactions (current value is {{ \$value }})
Summary	"timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 50 percent of total transactions"
Severity	Critical
Condition	The number of failed Fivegc Ingress transactions is above 50 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7018
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Fivegc Ingress transactions is below 50 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.1.1.19 OcnefEgressGatewayServiceDown

Table 7-19 OcnefEgressGatewayServiceDown

Field	Details
Description	"NEF Egress-Gateway service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query \"time()\" }}{ . first value humanizeTimestamp }}{ end }} : Egress-Gateway service down"
Severity	Critical
Condition	None of the pods of the Egress Gateway microservice is available.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7019
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the Egress Gateway service is available. Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of Egress Gateway service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> Refer to the application logs on Kibana and filter based on Egress Gateway service names. Check for ERROR WARNING logs related to thread exceptions. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

7.1.1.1.20 OcnefMemoryUsageCrossedMinorThreshold

Table 7-20 OcnefMemoryUsageCrossedMinorThreshold

Field	Details
Description	"NEF Memory Usage for pod {{ \$labels.pod }} has crossed the configured minor threshold (50%) (value={{ \$value }}) of its limit."
Summary	"namespace: {{ \$labels.namespace }}, podname: {{ \$labels.pod }}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 50% of its limit."
Severity	Minor
Condition	A pod has reached the configured minor threshold (50%) of its memory resource limits.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7020
Metric Used	'container_memory_usage_bytes'container_spec_memory_limit_bytes' Note: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the memory utilization falls below the Minor Threshold or crosses the major threshold, in which case OcnefMemoryUsageCrossedMajorThreshold alert is raised. Note: The threshold is configurable in the NefAlertrules alert file. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i> .

7.1.1.1.21 OcnefMemoryUsageCrossedMajorThreshold

Table 7-21 OcnefMemoryUsageCrossedMajorThreshold

Field	Details
Description	"NEF Memory Usage for pod {{ \$labels.pod }} has crossed the configured major threshold (60%) (value = {{ \$value }}) of its limit."
Summary	"namespace: {{ \$labels.namespace }}, podname: {{ \$labels.pod }}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 60% of its limit."
Severity	Major
Condition	A pod has reached the configured major threshold (60%) of its memory resource limits.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7021
Metric Used	'container_memory_usage_bytes' 'container_spec_memory_limit_bytes' Note: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.

Table 7-21 (Cont.) OcnefMemoryUsageCrossedMajorThreshold

Field	Details
Recommended Actions	<p>The alert gets cleared when the memory utilization falls below the Major Threshold or crosses the critical threshold, in which case OcnefMemoryUsageCrossedCriticalThreshold alert is raised.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</p> <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.1.1.22 OcnefMemoryUsageCrossedCriticalThreshold

Table 7-22 OcnefMemoryUsageCrossedCriticalThreshold

Field	Details
Description	"NEF Memory Usage for pod {{ \$labels.pod }} has crossed the configured major threshold (70%) (value = {{ \$value }}) of its limit."
Summary	"namespace: {{ \$labels.namespace }}, podname: {{ \$labels.pod }}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 70% of its limit."
Severity	Critical
Condition	A pod has reached the configured critical threshold (70%) of its memory resource limits.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7022
Metric Used	<p>'container_memory_usage_bytes'</p> <p>'container_spec_memory_limit_bytes'</p> <p>Note: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use a similar metric as exposed by the monitoring system.</p>
Recommended Actions	<p>The alert gets cleared when the memory utilization falls below the Critical threshold.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</p> <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.1.1.23 OcnfIngressGatewayServiceDown

Table 7-23 OcnfIngressGatewayServiceDown

Field	Details
Description	"NEF Ingress-Gateway service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Ingress-gateway service down"
Severity	Critical
Condition	None of the pods of the Ingress-Gateway microservice is available.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7023
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the Ingress Gateway service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of Ingress Gateway service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> Refer to the application logs on Kibana and filter based on Ingress Gateway service names. Check for ERROR WARNING logs related to thread exceptions. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

7.1.1.1.24 OcnefApiRouterServiceDown

Table 7-24 OcnefApiRouterServiceDown

Field	Details
Description	"NEF API Router service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"namespace: {{\$labels.namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : ApiRouter service down"
Severity	Critical
Condition	The API Router service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7024
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF API Router service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>ocnef_expgw_apirouter</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <code><pod name not in Running state></code> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>ocnef_expgw_apirouter</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, cnDBTier User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.1.1.25 OcnefFiveGcAgentServiceDown

Table 7-25 OcnefFiveGcAgentServiceDown

Field	Details
Description	"NEF FiveGc Agent service down {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : FiveGc Agent service down"
Severity	Critical
Condition	The 5GC Agent service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7025
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF 5GC Agent service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of 5gcagent service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on 5gcagent service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, cnDBTier User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.1.1.26 OcnefMonitoringEventServiceDown

Table 7-26 OcnefMonitoringEventServiceDown

Field	Details
Description	"NEF MonitoringEvent service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : MonitoringEvent service down"
Severity	Critical
Condition	The Monitoring Event (ME) service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7026
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF Monitoring Event (ME) service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>ocnef_monitoring_events</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <code><pod name not in Running state></code> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>ocnef_monitoring_events</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, <code>cnDBTier</code> User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.1.1.27 OcnfCCFClientServiceDown

Table 7-27 OcnfCCFClientServiceDown

Field	Details
Description	"NEF CCFClient service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : CCFClient service down"
Severity	Critical
Condition	The CCF Client service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7027
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF CCF Client service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>ocnef_ccfclient</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>ocnef_ccfclient</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, <code>cnDBTier</code> User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.1.1.28 OcnefExpiryAuditorServiceDown

Table 7-28 OcnefExpiryAuditorServiceDown

Field	Details
Description	"NEF Expiry Auditor service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Expiry Auditor service down"
Severity	Critical
Condition	The expiry auditor service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7028
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF Expiry Auditor service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>ocnef-expiry-auditor</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <code><pod name not in Running state></code> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>ocnef-expiry-auditor</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, <code>cnDBTier</code> User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.1.1.29 OcnefQOSServiceDown

Table 7-29 OcnefQOSServiceDown

Field	Details
Description	"NEF QoS service {{\$labels.app_kubernetes_io_name}} is down"
Summary	namespace: {{\$labels.namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : QoS service down
Severity	Critical
Condition	The QoS service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7029
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF QoS service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of ocnef-qualityofservice service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on ocnef-expiry-auditor service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, cnDBTier User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.1.1.30 OcnefTIServiceDown

Table 7-30 OcnefTIServiceDown

Field	Details
Description	OCNEF Traffic Influence service {{\$labels.app_kubernetes_io_name}} is down
Summary	namespace: {{\$labels.namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : TI service down
Severity	Critical
Condition	Traffic Influence service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7030
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF TI service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of ocnef-trafficinfluence service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on ocnef-expiry-auditor service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

7.1.1.1.31 OcnefDTServiceDown

Table 7-31 OcnefDTServiceDown

Field	Details
Description	OCNEF Device Trigger service {{\$labels.app_kubernetes_io_name}} is down
Summary	namespace: {{\$labels.namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : DT service down
Severity	Critical
Condition	Device Trigger service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7031
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the NEF DT service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>ocnef-devicetrigger</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <code><pod name not in Running state></code> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>ocnef-expiry-auditor</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

7.1.1.2 Application Level Alerts

This section lists the application level alerts for NEF.

7.1.1.2.1 AEFapiRouterOAuthValidationFailureRateCrossedThreshold

Table 7-32 AEFapiRouterOAuthValidationFailureRateCrossedThreshold

Field	Details
Description	"Failure Rate of API Router OAuth Validation Is Crossing the Threshold (10%)"
Summary	"{{ \$labels.namespace }}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : Failure Rate Of OAuth Validation is above 10 percent of total requests."
Severity	Error
Condition	The failure rate of the OAuth validations at API Router is reaching the threshold value.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7032
Metric Used	ocnef_aef_apirouter_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of OAuth validations at API Router is below the threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.2 MEAddSubscriptionFailureRateCrossedThreshold

Table 7-33 MEAddSubscriptionFailureRateCrossedThreshold

Field	Details
Description	"Failure Rate of ME Subscriptions Is Crossing the Threshold (10%)"
Summary	"namespace: {{ \$labels.namespace }}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : Failure Rate Of ME Subscriptions requests is above 10 percent of total requests."
Severity	Error
Condition	The failure rate of the Monitoring Event subscription requests is reaching the threshold value.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7033
Metric Used	ocnef_me_af_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of Monitoring Event subscription requests is below the threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.3 MDeleteSubscriptionFailureRateCrossedThreshold

Table 7-34 MDeleteSubscriptionFailureRateCrossedThreshold

Field	Details
Description	"Failure Rate of Delete ME Subscriptions Is Crossing the Threshold (10%)"
Summary	"namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Failure Rate Of delete ME Subscriptions requests is above 10 percent of total requests."
Severity	Error
Condition	The failure rate of the Monitoring Event subscription deletion requests is reaching the threshold value.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7034
Metric Used	ocnef_me_af_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of Monitoring Event subscription deletion requests is below the threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.4 MNotificationFailureRateCrossedThreshold

Table 7-35 MNotificationFailureRateCrossedThreshold

Field	Details
Description	"Failure Rate of Delete ME Notifications Is Crossing the Threshold (10%)"
Summary	"namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Failure Rate Of delete ME Subscriptions requests is above 10 percent of total requests."
Severity	Error
Condition	The failure rate of the DELETE Monitoring Event notification requests is reaching the threshold value.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7035
Metric Used	ocnef_me_af_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of DELETE ME notification requests is below the threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.5 FiveGcInvalidConfiguration

Table 7-36 FiveGcInvalidConfiguration

Field	Details
Description	"Invalid Configuration For Five GC Service"
Summary	"namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Invalid Configuration For Five GC Service."
Severity	Error
Condition	Invalid configuration of the 5GCAgent service.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7036
Metric Used	ocnef_5gc_invalid_config
Recommended Actions	<p>The alert is cleared when the 5GCAgent service configuration are valid.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.6 QOSAddSubscriptionFailureRateCrossedThreshold

Table 7-37 QOSAddSubscriptionFailureRateCrossedThreshold

Field	Details
Description	Failure rate of QoS subscriptions is crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}
Severity	Error
Condition	Failure rate of QoS subscription requests is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7037
Metric Used	ocnef_qos_af_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of subscription requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.7 QOSDeleteSubscriptionFailureRateCrossedThreshold

Table 7-38 QOSDeleteSubscriptionFailureRateCrossedThreshold

Field	Details
Description	Failure rate of delete QoS subscriptions is crossing the threshold (10%).

Table 7-38 (Cont.) QOSDeleteSubscriptionFailureRateCrossedThreshold

Field	Details
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	Error
Condition	Failure rate of delete QoS subscriptions requests is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7038
Metric Used	ocnef_qos_af_resp_total
Recommended Actions	The alert is cleared when the failure rate of subscription requests is below the failure threshold. Steps: <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.8 QOSNotificationFailureRateCrossedThreshold

Table 7-39 QOSNotificationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of QoS notifications is crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	Error
Condition	Failure rate of QoS notifications requests is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7039
Metric Used	ocnef_qos_5g_resp_total
Recommended Actions	The alert is cleared when the failure rate of notification requests is below the failure threshold. Steps: <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.9 OcnefAllSiteStatus

Table 7-40 OcnefAllSiteStatus

Field	Details
Description	"Alert for any NEF sites status if SUSPENDED in Georedundant setup"
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} : Alert for any NEF sites status if SUSPENDED in Georedundant setup
Severity	Error

Table 7-40 (Cont.) OcnefAllSiteStatus

Field	Details
Condition	An NEF site of a georedundant deployment is in Suspended state.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7040
Metric Used	ocnef_all_site_status
Recommended Actions	<p>The alert is cleared when all the sites in a georedundant deployment are UP.</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.10 OcnefDBReplicationStatus

Table 7-41 OcnefDBReplicationStatus

Field	Details
Description	"Alert for NEF sites status if DB Replication down in Georedundant setup"
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} : Alert for NEF sites status if DB Replication down in Georedundant setup
Severity	Error
Condition	The database replication channel status between the given site and the georedundant site(s) is inactive. The alert is raised per replication channel. The alarm is raised or cleared only if the georedundancy feature is enabled.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7041
Metric Used	ocnef_db_replication_status
Recommended Actions	The alert is cleared when the database channel replication status between the given site and the georedundant site(s) is UP. For more information on how to check the database replication status, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i> .

7.1.1.2.11 MeEPCAddSubscriptionFailureRateCrossedThreshold

Table 7-42 MeEPCAddSubscriptionFailureRateCrossedThreshold

Field	Details
Description	Failure rate of ME subscriptions to EPC crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure Rate of ME EPC Subscriptions requests is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7042
Metric Used	ocnef_me_epc_sub_total

Table 7-42 (Cont.) MeEPCAddSubscriptionFailureRateCrossedThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.12 DiameterGwT6InvocationFailureRateCrossedThreshold

Table 7-43 DiameterGwT6InvocationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway T6x Invocation requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway T6x Invocation requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7043
Metric Used	ocnef_diamgw_diam_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.13 DiameterGwT4InvocationFailureRateCrossedThreshold

Table 7-44 DiameterGwT4InvocationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway T4 Invocation requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway T4 Invocation requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7044
Metric Used	ocnef_diamgw_diam_resp_total

Table 7-44 (Cont.) DiameterGwT4InvocationFailureRateCrossedThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.14 DiameterGwRxInvocationFailureRateCrossedThreshold

Table 7-45 DiameterGwRxInvocationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway Rx Invocation requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway Rx Invocation requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7045
Metric Used	ocnef_diamgw_diam_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.15 DiameterGwSgdT4InvocationFailureRateCrossedThreshold

Table 7-46 DiameterGwSgdT4InvocationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway SgdT4 Invocation requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway SgdT4 Invocation requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7046
Metric Used	ocnef_diamgw_diam_resp_total

Table 7-46 (Cont.) DiameterGwSgdT4InvocationFailureRateCrossedThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.16 DiameterGwT6NotificationFailureRateCrossedThreshold

Table 7-47 DiameterGwT6NotificationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway T6x Notification requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway T6x Notification requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7047
Metric Used	ocnef_diamgw_diam_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.17 DiameterGwT4NotificationFailureRateCrossedThreshold

Table 7-48 DiameterGwT4NotificationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway T4 Notification requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway T4 Notification requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7048
Metric Used	ocnef_diamgw_diam_resp_total

Table 7-48 (Cont.) DiameterGwT4NotificationFailureRateCrossedThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.18 DiameterGwRxNotificationFailureRateCrossedThreshold

Table 7-49 DiameterGwRxNotificationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway Rx Notification requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway Rx Notification requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7049
Metric Used	ocnef_diamgw_diam_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.19 DiameterGwSgdT4NotificationFailureRateCrossedThreshold

Table 7-50 DiameterGwSgdT4NotificationFailureRateCrossedThreshold

Field	Details
Description	Failure rate of Diameter Gateway SgdT4 Notification requests crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of Diameter Gateway SgdT4 Notification requests crossing the threshold (10%).
OID	1.3.6.1.4.1.323.5.3.39.1.2.7050
Metric Used	ocnef_diamgw_diam_resp_total

Table 7-50 (Cont.) DiameterGwSgdT4NotificationFailureRateCrossedThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.20 DiameterGwT6TranslationFailureRateCrossedThreshold

Table 7-51 DiameterGwT6TranslationFailureRateCrossedThreshold

Field	Details
Description	Failure Rate of T6x Translations In Diameter GW is above 10 percent of total requests.
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure Rate of T6x Translations In Diameter GW is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7051
Metric Used	ocnef_diamgw_translator_request_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.21 DiameterGwT4TranslationFailureRateCrossedThreshold

Table 7-52 DiameterGwT4TranslationFailureRateCrossedThreshold

Field	Details
Description	Failure Rate of T4 Translations In Diameter GW is above 10 percent of total requests.
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure Rate of T4 Translations In Diameter GW is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7052
Metric Used	ocnef_diamgw_translator_request_total

Table 7-52 (Cont.) DiameterGwT4TranslationFailureRateCrossedThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.22 DiameterGwRxTranslationFailureRateCrossedThreshold

Table 7-53 DiameterGwRxTranslationFailureRateCrossedThreshold

Field	Details
Description	Failure Rate of Rx Translations In Diameter GW is above 10 percent of total requests.
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure Rate of Rx Translations In Diameter GW is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7053
Metric Used	ocnef_diamgw_translator_request_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.23 DiameterGwSgdT4TranslationFailureRateCrossedThreshold

Table 7-54 DiameterGwSgdT4TranslationFailureRateCrossedThreshold

Field	Details
Description	Failure Rate of SgdT4 Translations in Diameter GW is above 10 percent of total requests.
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure Rate of SgdT4 Translations in Diameter GW is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7054
Metric Used	ocnef_diamgw_translator_request_total

Table 7-54 (Cont.) DiameterGwSgdT4TranslationFailureRateCrossedThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.24 CHFAddChargingDataRequestFailureRateCrossedErrorThreshold

Table 7-55 CHFAddChargingDataRequestFailureRateCrossedErrorThreshold

Field	Details
Description	Failure rate of CHF Create Charging Data request is crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	error
Condition	Failure rate of CHF Create Charging Data request is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7055
Metric Used	ocnef_chf_qos_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.25 CHFAddChargingDataRequestFailureRateCrossedCriticalThreshold

Table 7-56 CHFAddChargingDataRequestFailureRateCrossedCriticalThreshold

Field	Details
Description	Failure rate of CHF Create Charging Data request is crossing the threshold (25%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	critical
Condition	Failure rate of CHF Create Charging Data request is above 25 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7056
Metric Used	ocnef_chf_qos_resp_total

Table 7-56 (Cont.) CHFAddChargingDataRequestFailureRateCrossedCriticalThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.26 CHFAddChargingDataRequestFailureRateCrossedMinorThreshold

Table 7-57 CHFAddChargingDataRequestFailureRateCrossedMinorThreshold

Field	Details
Description	Failure rate of CHF Create Charging Data request is crossing the threshold (5%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	minor
Condition	Failure rate of CHF Create Charging Data request is above 5 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7057
Metric Used	ocnef_chf_qos_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of CHF Requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.27 MSISDNLessMoSMSRequestFailureRateCrossedCriticalThreshold

Table 7-58 MSISDNLessMoSMSRequestFailureRateCrossedCriticalThreshold

Field	Details
Description	Failure rate of MSISDNLess MO SMS notification request is crossing the threshold (25%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	Critical
Condition	Failure rate Of MSISDNLess MO SMS request is above 25 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7058
Metric Used	ocnef_msisdnless_mo_sms_diamgw_notify_resp_total

Table 7-58 (Cont.) MSISDNLessMoSMSRequestFailureRateCrossedCriticalThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of notification requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.28 MSISDNLessMoSMSRequestFailureRateCrossedMajorThreshold

Table 7-59 MSISDNLessMoSMSRequestFailureRateCrossedMajorThreshold

Field	Details
Description	Failure rate of MSISDNLess MO SMS notification request is crossing the threshold (10%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	Major
Condition	Failure rate of MSISDNLess MO SMS request is above 10 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7059
Metric Used	ocnef_msisdnless_mo_sms_diamgw_notify_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of notification requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.29 MSISDNLessMoSMSRequestFailureRateCrossedMinorThreshold

Table 7-60 MSISDNLessMoSMSRequestFailureRateCrossedMinorThreshold

Field	Details
Description	Failure rate of MSISDNLess MO SMS notification request is crossing the threshold (5%).
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	Minor
Condition	Failure rate of MSISDNLess MO SMS request is above 5 percent of total requests.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7060
Metric Used	ocnef_msisdnless_mo_sms_diamgw_notify_resp_total

Table 7-60 (Cont.) MSISDNLessMoSMSRequestFailureRateCrossedMinorThreshold

Field	Details
Recommended Actions	<p>The alert is cleared when the failure rate of notification requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.1.2.30 MSISDNLessMoSMSShortCodeConfigMatchFailure

Table 7-61 MSISDNLessMoSMSShortCodeConfigMatchFailure

Field	Details
Description	Failure when shortcode configured doesn't match the shortcode from incoming request.
Summary	namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}
Severity	Error
Condition	Failure when shortcode configured doesn't match the shortcode from SMSSC.
OID	1.3.6.1.4.1.323.5.3.39.1.2.7061
Metric Used	ocnef_diamgw_http_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of notification requests is below the failure threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.1.2 CAPIF Alerts

This chapter includes information about the following CAPIF alerts:

- [System Level Alerts](#)
- [Application Level Alerts](#)

Note

- The performance and capacity of the CAPIF system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.
- Due to unavailability of metric and/or MQL queries, the following alerts are not supported for OCI:
 - OccapifNfStatusUnavailable
 - OccapifPodsRestart
 - OccapifEgressGatewayServiceDown
 - OccapifIngressGatewayServiceDown
 - OccapifAfManagerServiceDown
 - OccapifAPIManagerServiceDown
 - OccapifEventManagerServiceDown

7.1.2.1 System Level Alerts

This section lists the system level alerts for CAPIF.

7.1.2.1.1 OccapifNfStatusUnavailable

Table 7-62 OccapifNfStatusUnavailable

Field	Details
Description	CAPIF services unavailable'
Summary	"namespace: {{\$labels.namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : All OCCAPIF services are unavailable."
Severity	Critical
Condition	All the CAPIF services are unavailable.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5001
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.

Table 7-62 (Cont.) OccapifNfStatusUnavailable

Field	Details
Recommended Actions	<p>The alert is cleared automatically when the CAPIF services restart.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for service-specific alerts which may be causing the issues with service exposure. 2. Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> <ol style="list-style-type: none"> a. Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> 3. Refer to the application logs on Kibana and check for database related failures such as connectivity and invalid secrets. The logs can be filtered based on the services. 4. Check for helm status to make sure there are no errors: <pre>\$ helm status <helm release name of the desired NF> -n <namespace></pre> <p>If it is not in "STATUS : DEPLOYED", then capture logs and event again.</p> 5. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support. <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on the Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.2.1.2 OccapifPodsRestart

Table 7-63 OccapifPodsRestart

Field	Details
Description	'Pod <Pod Name> has restarted.
Summary	"namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : A Pod has restarted"
Severity	Major
Condition	A pod belonging to any of the CAPIF services has restarted.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5002
Metric Used	kube_pod_container_status_restarts_total

Table 7-63 (Cont.) OccapifPodsRestart

Field	Details
Recommended Actions	<p>The alert is cleared automatically if the specific pod is up.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer to the application logs on Kibana and filter based on pod name, check for database related failures such as connectivity and Kubernetes secrets. 2. To check the orchestration logs for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> a. Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> b. Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support. <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on the Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.2.1.3 OccapifTotalExternalIngressTrafficRateAboveMinorThreshold

Table 7-64 OccapifTotalExternalIngressTrafficRateAboveMinorThreshold

Field	Details
Description	"OCCAPIF External Ingress traffic rate is above the configured minor threshold i.e. 800 TPS (current value is: {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 80 percent of max TPS (1000)"
Severity	Minor
Condition	The total CAPIF External Ingress traffic rate has crossed the configured minor threshold of 800 TPS. Default value of this alert trigger point in <i>Occapif Alert.yaml</i> is 80 % of 1000 (Maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.39.1.3.5003
Metric Used	oc_ingressgateway_http_requests_total

Table 7-64 (Cont.) OccapifTotalExternalIngressTrafficRateAboveMinorThreshold

Field	Details
Recommended Actions	<p>The alert is cleared either when the External Ingress traffic rate goes above the minor threshold.</p> <p>Note: The threshold is configurable in the <i>Occapif Alert.yaml</i> alert file. Reassess why the CAPIF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.2.1.4 OccapifTotalNetworkIngressTrafficRateAboveMinorThreshold

Table 7-65 OccapifTotalNetworkIngressTrafficRateAboveMinorThreshold

Field	Details
Description	"OCCAPIF Network Ingress traffic rate is above the configured minor threshold i.e. 800 TPS (current value is: {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 80 percent of max TPS (1000)"
Severity	Minor
Condition	<p>The total CAPIF Network Ingress traffic rate has crossed the configured minor threshold of 800 TPS.</p> <p>Default value of this alert trigger point in <i>Occapif Alert.yaml</i> is 80% of 1000 (maximum ingress request rate).</p>
OID	1.3.6.1.4.1.323.5.3.39.1.3.5004
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the Network Ingress traffic rate goes above the minor threshold.</p> <p>Note: The threshold is configurable in the <i>Occapif Alert.yaml</i> alert file. Reassess why the CAPIF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.2.1.5 OccapifTotalExternalIngressTrafficRateAboveMajorThreshold

Table 7-66 OccapifTotalExternalIngressTrafficRateAboveMajorThreshold

Field	Details
Description	"OCCAPIF External Ingress traffic rate is above the configured major threshold i.e. 900 TPS (current value is: {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 90 percent of max TPS (1000)"
Severity	Major
Condition	The total CAPIF External Ingress traffic rate has crossed the configured major threshold of 900 TPS. Default value of this alert trigger point in <i>Occapif Alert.yaml</i> is 90 % of 1000 (maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.39.1.3.5005
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the External Ingress traffic rate goes above the major threshold.</p> <p>Note: The threshold is configurable in the <i>Occapif Alert.yaml</i> alert file. Reassess why the CAPIF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.2.1.6 OccapifTotalNetworkIngressTrafficRateAboveMajorThreshold

Table 7-67 OccapifTotalNetworkIngressTrafficRateAboveMajorThreshold

Field	Details
Description	"OCCAPIF Network Ingress traffic rate is above the configured major threshold i.e. 900 TPS (current value is: {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 percent of max TPS (1000)"
Severity	Major
Condition	The total CAPIF Network Ingress traffic rate has crossed the configured major threshold of 900 TPS. Default value of this alert trigger point in <i>Occapif Alert.yaml</i> is 90 % of 1000 (maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.39.1.3.5006
Metric Used	oc_ingressgateway_http_requests_total

Table 7-67 (Cont.) OccapifTotalNetworkIngressTrafficRateAboveMajorThreshold

Field	Details
Recommended Actions	<p>The alert is cleared either when the Network Ingress traffic rate goes above the major threshold.</p> <p>Note: The threshold is configurable in the <i>Occapif Alert.yaml</i> alert file. Reassess why the CAPIF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.2.1.7 OccapifTotalExternalIngressTrafficRateAboveCriticalThreshold

Table 7-68 OccapifTotalExternalIngressTrafficRateAboveCriticalThreshold

Field	Details
Description	"OCCAPIF External Ingress traffic rate is above the configured critical threshold i.e. 950 TPS (current value is: {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 95 percent of max TPS (1000)"
Severity	Critical
Condition	<p>The total CAPIF External Ingress traffic rate has crossed the configured critical threshold of 950 TPS.</p> <p>Default value of this alert trigger point in <i>Occapif Alert.yaml</i> is 95 % of 1000 (maximum ingress request rate).</p>
OID	1.3.6.1.4.1.323.5.3.39.1.3.5007
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the External Ingress traffic rate goes above the critical threshold.</p> <p>Note: The threshold is configurable in the <i>Occapif Alert.yaml</i> alert file. Reassess why the CAPIF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.2.1.8 OccapifTotalNetworkIngressTrafficRateAboveCriticalThreshold

Table 7-69 OccapifTotalNetworkIngressTrafficRateAboveCriticalThreshold

Field	Details
Description	"OCCAPIF Network Ingress traffic rate is above the configured critical threshold i.e. 950 TPS (current value is: {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Traffic rate is above 95 percent of max TPS (1000)"
Severity	Critical
Condition	The total CAPIF Network Ingress traffic rate has crossed the configured critical threshold of 950 TPS. Default value of this alert trigger point in <i>Occapif.Alert.yaml</i> is 95 % of 1000 (Maximum ingress request rate).
OID	1.3.6.1.4.1.323.5.3.39.1.3.5008
Metric Used	oc_ingressgateway_http_requests_total
Recommended Actions	<p>The alert is cleared either when the Network Ingress traffic rate goes above the critical threshold.</p> <p>Note: The threshold is configurable in the <i>Occapif.Alert.yaml</i> alert file. Reassess why the CAPIF is receiving additional traffic. If this alert is unexpected, contact My Oracle Support.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Refer Grafana to determine which service is receiving high traffic. 2. Refer Ingress gateway section in Grafana to determine the increase in 4xx and 5xx error codes. 3. Check Ingress gateway logs on Kibana to determine the reason for the errors.

7.1.2.1.9 OccapifExternalIngressTransactionErrorRateAboveZeroPointOnePercent

Table 7-70 OccapifExternalIngressTransactionErrorRateAboveZeroPointOnePercent

Field	Details
Description	"OCCAPIF External Ingress transaction error rate is above 0.1 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 0.1 percent of total transactions"
Severity	Warning
Condition	The number of failed External Ingress transactions is above 0.1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5009
Metric Used	oc_ingressgateway_http_responses_total

Table 7-70 (Cont.)
OccapifExternalIngressTransactionErrorRateAboveZeroPointOnePercent

Field	Details
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress 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 OccapifExternalIngressTransactionErrorRateAbove1Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.10 OccapifNetworkIngressTransactionErrorRateAboveZeroPointOnePercent

Table 7-71 **OccapifNetworkIngressTransactionErrorRateAboveZeroPointOnePercent**

Field	Details
Description	"OCCAPIF Network Ingress transaction error rate is above 0.1 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{ . first value humanizeTimestamp }}{ end }}: Transaction error rate detected above 0.1 percent of total transactions"
Severity	Warning
Condition	The number of failed Network Ingress transactions is above 0.1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5010
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Network Ingress 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 OccapifNetworkIngressTransactionErrorRateAbove1Percent is raised.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.11 OccapifExternalIngressTransactionErrorRateAbove1Percent

Table 7-72 **OccapifExternalIngressTransactionErrorRateAbove1Percent**

Field	Details
Description	"OCCAPIF External Ingress transaction error rate is above 1 percent of total transactions (current value is {{ \$value }})"

Table 7-72 (Cont.) OccapifExternalIngressTransactionErrorRateAbove1Percent

Field	Details
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 1 percent of total transactions"
Severity	Warning
Condition	The number of failed External Ingress transactions is above 1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5011
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 1 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.12 OccapifNetworkIngressTransactionErrorRateAbove1Percent

Table 7-73 OccapifNetworkIngressTransactionErrorRateAbove1Percent

Field	Details
Description	"OCCAPIF Network Ingress transaction error rate is above 1 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 1 percent of total transactions"
Severity	Warning
Condition	The number of failed Network Ingress transactions is above 1 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5012
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Network Ingress transactions is below 1 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.13 OccapifExternalIngressTransactionErrorRateAbove10Percent

Table 7-74 OccapifExternalIngressTransactionErrorRateAbove10Percent

Field	Details
Description	"OCCAPIF External Ingress transaction error rate is above 10 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 10 percent of total transactions"
Severity	Minor
Condition	The number of failed External Ingress transactions is above 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5013
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 10 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.14 OccapifNetworkIngressTransactionErrorRateAbove10Percent

Table 7-75 OccapifNetworkIngressTransactionErrorRateAbove10Percent

Field	Details
Description	"OCCAPIF Network Ingress transaction error rate is above 10 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 10 percent of total transactions"
Severity	Minor
Condition	The number of failed Network Ingress transactions is above 10 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5014
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Network Ingress transactions is below 10 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.15 OccapifExternalIngressTransactionErrorRateAbove25Percent

Table 7-76 OccapifExternalIngressTransactionErrorRateAbove25Percent

Field	Details
Description	"OCCAPIF External Ingress transaction error rate detected above 25 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 25 percent of total transactions"
Severity	Major
Condition	The number of failed External Ingress transactions is above 25 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5015
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 25 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.16 OccapifNetworkIngressTransactionErrorRateAbove25Percent

Table 7-77 OccapifNetworkIngressTransactionErrorRateAbove25Percent

Field	Details
Description	"OCCAPIF Network Ingress transaction error rate detected above 25 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 25 percent of total transactions"
Severity	Major
Condition	The number of failed Network Ingress transactions is above 25 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5016
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Network Ingress transactions is below 25 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.17 OccapifExternalIngressTransactionErrorRateAbove50Percent

Table 7-78 OccapifExternalIngressTransactionErrorRateAbove50Percent

Field	Details
Description	"OCCAPIF External Ingress transaction error rate detected above 50 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 50 percent of total transactions"
Severity	Critical
Condition	The number of failed External Ingress transactions is above 50 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5017
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure External Ingress transactions is below 50 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.18 OccapifNetworkIngressTransactionErrorRateAbove50Percent

Table 7-79 OccapifNetworkIngressTransactionErrorRateAbove50Percent

Field	Details
Description	"OCCAPIF Network Ingress transaction error rate detected above 50 percent of total transactions (current value is {{ \$value }})"
Summary	"timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }}: Transaction error rate detected above 50 percent of total transactions"
Severity	Critical
Condition	The number of failed Network Ingress transactions is above 50 percent of the total transactions.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5018
Metric Used	oc_ingressgateway_http_responses_total
Recommended Actions	<p>The alert is cleared when the number of failure Network Ingress transactions is below 50 percent of the total transactions.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check the service specific metrics to understand the specific service request errors. 2. Check metrics per service, per method: 3. If guidance is required, contact My Oracle Support.

7.1.2.1.19 OccapifEgressGatewayServiceDown

Table 7-80 OccapifEgressGatewayServiceDown

Field	Details
Description	"CAPIF Egress-Gateway service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query \"time()\" }}{ . first value humanizeTimestamp }}{ end } : Egress-Gateway service down"
Severity	Critical
Condition	None of the pods of the Egress Gateway microservice is available.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5019
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the Egress Gateway service is available. Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of Egress Gateway service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> Refer to the application logs on Kibana and filter based on Egress Gateway service names. Check for ERROR WARNING logs related to thread exceptions. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

7.1.2.1.20 OccapifMemoryUsageCrossedMinorThreshold

Table 7-81 OccapifMemoryUsageCrossedMinorThreshold

Field	Details
Description	"CAPIF Memory Usage for pod {{ \$labels.pod }} has crossed the configured minor threshold (50%) (value={{ \$value }}) of its limit."
Summary	"namespace: {{ \$labels.namespace }}, podname: {{ \$labels.pod }}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 50% of its limit."
Severity	Minor
Condition	A pod has reached the configured minor threshold (50%) of its memory resource limits.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5020
Metric Used	'container_memory_usage_bytes'container_spec_memory_limit_bytes' Note: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.
Recommended Actions	The alert gets cleared when the memory utilization falls below the Minor Threshold or crosses the major threshold, in which case OccapifMemoryUsageCrossedMajorThreshold alert is raised. Note: The threshold is configurable in the NefAlertrules alert file. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i> .

7.1.2.1.21 OccapifMemoryUsageCrossedMajorThreshold

Table 7-82 OccapifMemoryUsageCrossedMajorThreshold

Field	Details
Description	"CAPIF Memory Usage for pod {{ \$labels.pod }} has crossed the configured major threshold (60%) (value = {{ \$value }}) of its limit."
Summary	"namespace: {{ \$labels.namespace }}, podname: {{ \$labels.pod }}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 60% of its limit."
Severity	Major
Condition	A pod has reached the configured major threshold (60%) of its memory resource limits.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5021
Metric Used	'container_memory_usage_bytes' 'container_spec_memory_limit_bytes' Note: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.

Table 7-82 (Cont.) OccapifMemoryUsageCrossedMajorThreshold

Field	Details
Recommended Actions	<p>The alert gets cleared when the memory utilization falls below the Major Threshold or crosses the critical threshold, in which case OccapifMemoryUsageCrossedCriticalThreshold alert is raised.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</p> <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.2.1.22 OccapifMemoryUsageCrossedCriticalThreshold

Table 7-83 OccapifMemoryUsageCrossedCriticalThreshold

Field	Details
Description	"CAPIF Memory Usage for pod {{ \$labels.pod }} has crossed the configured major threshold (70%) (value = {{ \$value }}) of its limit."
Summary	"namespace: {{ \$labels.namespace }}, podname: {{ \$labels.pod }}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 70% of its limit."
Severity	Critical
Condition	A pod has reached the configured critical threshold (70%) of its memory resource limits.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5022
Metric Used	<p>'container_memory_usage_bytes'</p> <p>'container_spec_memory_limit_bytes'</p> <p>Note: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use a similar metric as exposed by the monitoring system.</p>
Recommended Actions	<p>The alert gets cleared when the memory utilization falls below the Critical threshold.</p> <p>Note: The threshold is configurable in the NefAlertrules alert file.</p> <p>In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.</p> <p>Note: Use CNC NF Data Collector tool for capturing logs. For more information on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.</p>

7.1.2.1.23 OccapifIngressGatewayServiceDown

Table 7-84 OccapifIngressGatewayServiceDown

Field	Details
Description	"CAPIF Ingress-Gateway service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{ . first value humanizeTimestamp }}{ end }} : Ingress-gateway service down"
Severity	Critical
Condition	None of the pods of the Ingress-Gateway microservice is available.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5023
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the Ingress Gateway service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of Ingress Gateway service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get po -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the <i>Running</i> state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <pod name not in Running state> indicates the pod that is not in the <i>Running</i> state.</p> Refer to the application logs on Kibana and filter based on Ingress Gateway service names. Check for ERROR WARNING logs related to thread exceptions. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see <i>Oracle Communications Cloud Native Core, Network Function Data Collector User Guide</i>.

7.1.2.1.24 OcapifAfManagerServiceDown

Table 7-85 OcapifAfManagerServiceDown

Field	Details
Description	"CAPIF AF Manager service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{ . first value humanizeTimestamp }}{ end }} : AF Manager service down"
Severity	Critical
Condition	The AF Manager service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5024
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the CAPIF AF Manager service is available. Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>occapif_afmgr</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <code><pod name not in Running state></code> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>occapif_afmgr</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, cnDBTier User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.2.1.25 OccapifApiManagerServiceDown

Table 7-86 OccapifApiManagerServiceDown

Field	Details
Description	"CAPIF API Manager service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"namespace: {{\$labels.namespace}}, podname: {{\$labels.kubernetes_pod_name}}, timestamp: {{ with query \"time()\" }} {{ . first value humanizeTimestamp }}{{ end }} : AF Manager service down"
Severity	Critical
Condition	The API Manager service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5025
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the CAPIF API Manager service is available. Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>occapif_apimgr</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <code><pod name not in Running state></code> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>occapif_apimgr</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, <code>cnDBTier</code> User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.2.1.26 OccapifEventManagerServiceDown

Table 7-87 OccapifEventManagerServiceDown

Field	Details
Description	"CAPIF API Manager service {{\$labels.app_kubernetes_io_name}} is down"
Summary	"kubernetes_namespace: {{\$labels.kubernetes_namespace}}, timestamp: {{ with query \"time()\" }}{ . first value humanizeTimestamp }}{ end }} : API Manager service down"
Severity	Critical
Condition	The Event Manager service is down.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5026
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Recommended Actions	<p>The alert is cleared when the CAPIF Event Manager service is available.</p> <p>Steps:</p> <ol style="list-style-type: none"> To check the orchestration logs of <code>occapif_eventmanager</code> service and check for liveness or readiness probe failures, do the following: <ol style="list-style-type: none"> Run the following command to check the pod status: <pre>\$ kubectl get pod -n <namespace></pre> Run the following command to analyze the error condition of the pod that is not in the Running state: <pre>\$ kubectl describe pod <pod name not in Running state> -n <namespace></pre> <p>Where <code><pod name not in Running state></code> indicates the pod that is not in the Running state.</p> Refer the application logs on Kibana and filter based on <code>ocnef_expgw_apimgr</code> service names. Check for ERROR WARNING logs related to thread exceptions. Check the DB status. For more information on how to check the DB status, see Oracle Communications Cloud Native Core, <code>cnDBTier</code> User Guide. Depending on the failure reason, take the resolution steps. 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 on how to collect logs using Data Collector tool, see Oracle Communications Cloud Native Core, Network Function Data Collector User Guide.

7.1.2.2 Application Level Alerts

This section lists the application level alerts for CAPIF.

7.1.2.2.1 AfMgrOnboardingOauthValidationFailureRateCrossedThreshold

Table 7-88 AfMgrOnboardingOauthValidationFailureRateCrossedThreshold

Field	Details
Description	"Failure Rate of AI Onboarding Oauth Validation Is Crossing the Threshold (10%)"
Summary	"namespace: {{\$labels.namespace}}, timestamp: {{ with query \"time()\" }}{{ . first value humanizeTimestamp }}{{ end }} : Failure Rate Of Onboarding is above 10 percent of total requests."
Severity	Error
Condition	The failure rate of API Invoker onboarding is reaching the threshold value.
OID	1.3.6.1.4.1.323.5.3.39.1.3.5027
Metric Used	occapif_afmgr_resp_total
Recommended Actions	<p>The alert is cleared when the failure rate of API invoker onboarding is below the threshold.</p> <p>Steps:</p> <ol style="list-style-type: none"> 1. Check for pod logs on Kibana for ERROR WARN logs. 2. In case the issue persists, capture all the outputs for the above steps and contact My Oracle Support.

7.2 Configuring Alerts

This section describes the measurement based alert rules configuration for NEF and CAPIF. The Alert Manager uses the Prometheus measurements values as reported by microservices in conditions under alert rules to trigger alerts.

The NEF package contains the sample alert files with the NEF Custom Templates. The NEF Custom Templates.zip file can be downloaded from MOS. Unzip the folder to access the following files:

- NefAlertrules-24.2.2.yaml
- CapifAlertrules-24.2.2.yaml

Note

- If required, edit the threshold values of various alerts in the alert files before configuring the alerts. For more information on the alerts that can be updated, see [Alert Details](#).
- The Alert Manager and Prometheus tools must run in Oracle CNE namespace, for example, occne-infra.

Table 7-89 Alert Details

Alert Name	Details	Default Value	Notes
OcnefTotalIngressTrafficRateAboveMinorThreshold	Traffic Rate is above 80 Percent of Max requests per second	Greater than/equal to 800 and Less than 900	Maximum Ingress rate considered is 1000 requests per second. So, here in default value 800 is 80% of 1000 and 900 is 90% of 1000. For example, if value need to be updated then depending upon maximum ingress request rate, set [90% of Max Ingress Request Rate] and [80% of Max Ingress Request Rate] for this alert
OcnefTotalIngressTrafficRateAboveMajorThreshold	Traffic Rate is above 90 Percent of Max requests per second	Greater than/equal to 900 and Less than 950	Maximum Ingress rate considered is 1000 requests per second. So, here in default value 900 is 90% of 1000 and 950 is 95% of 1000. For example, if value need to be updated then depending upon maximum ingress request rate, set [90% of Max Ingress Request Rate] and [95% of Max Ingress Request Rate] for this alert
OcnefTotalIngressTrafficRateAboveCriticalThreshold	Traffic Rate is above 95 Percent of Max requests per second	Greater than/equal to 950	Maximum Ingress rate considered is 1000 requests per second. So, here in default value 950 is 95% of 1000. For example, if value need to be updated then depending upon maximum ingress request rate, set [95% of Max Ingress Request Rate] for this alert

Update NEF alerts for OCCNE 1.8.x and previous versions

The following procedure describes how to update the NEF alerts for OCCNE version 1.8.x and previous versions:

NAME :- Helm Release of Prometheus

Namespace :- Kubernetes NameSpace in which Prometheus is installed

1. Find the config map to configure alerts in Prometheus server using the following command:

```
kubectl get configmap -n <Namespace>
```

where, <Namespace> is the prometheus server namespace used in the helm install command. For example, assuming Prometheus server is under occne-infra namespace, run the following command to find the config map:

```
kubectl get configmaps -n occne-infra | grep prometheus-server
```

2. Take backup of current configuration map of Prometheus:

```
kubectl get configmaps _NAME_-server -o yaml -n _Namespace_ > /tmp/tempConfig.yaml
```


3. Check if **alertsnef** is present in the **tempConfig.yaml** file by running the following command:

```
cat /tmp/tempConfig.yaml | grep alertsnef
```

Depending on the outcome of the previous step, perform anyone of the following steps:

- a. If **alertsnef** is present, delete the **alertsnef** entry from the **tempConfig.yaml** file, by running the following commands:

```
sed -i '/etc\/config\/alertsnef/d' /tmp/tempConfig.yaml
```

```
sed -i '/rule_files:/a\ \- /etc/config/alertsnef' /tmp/tempConfig.yaml
```

Note

This command should be run only once.

- b. If **alertsnef** is not present, add the **alertsnef** entry in the **tempConfig.yaml** file by running the following command:

```
sed -i '/rule_files:/a\ \- /etc/config/alertsnef' /tmp/tempConfig.yaml
```

4. Update configuration map with updated file name of NEF alert file:

```
kubectl replace configmap _NAME_-server -f /tmp/tempConfig.yaml
```

5. Add NEF alert rules in configuration map under file name of NEF alert file:

```
kubectl patch configmap _NAME_-server -n _Namespace_--type merge --patch
"$(cat ~/NefAlertrules-24.2.2.yaml)"
```

Update CAPIF/NEF alerts for OCCNE 1.9.x and later

This section describes the measurement based Alert rules configuration for CAPIF/NEF in Prometheus. Use the `NefAlertrules-24.2.2.yaml` or `CapifAlertrules-24.2.2.yaml` file updated in Alert configuration section.

1. Run the following command to apply the prometheusrules CRD:

```
$ kubectl apply -f <alert.yaml file> --namespace <namespace>
```

Example for NEF:

```
$ kubectl apply -f NefAlertrules-24.2.2.yaml --namespace ocnef
prometheusrule.monitoring.coreos.com/nef-alerting-rules created
```

Example for CAPIF:

```
$ kubectl apply -f CapifAlertrules-24.2.2.yaml --namespace ocnef
prometheusrule.monitoring.coreos.com/capif-alerting-rules created
```


2. Run the following command to check CAPIF/NEF alert file is added to prometheusrules:

```
$ kubectl get prometheusrules --namespace <namespace>
```

Example for NEF:

```
$ kubectl get prometheusrules --namespace nef
```

Sample output:

NAME	AGE
nef-alerting-rules	1m

Example for CAPIF:

```
$ kubectl get prometheusrules --namespace capif
```

Sample output:

NAME	AGE
capif-alerting-rules	1m

3. Log in to Prometheus GUI and verify the alerts section.

Note

The Prometheus server takes an updated configuration map that is automatically reloaded after approximately 60 seconds. Refresh the Prometheus GUI to confirm that the CAPIF/NEF Alerts have been reloaded.

7.2.1 Configuring Alert Manager for SNMP Notifier

This section describes the procedure to configure SNMP Notifier.

Configure the IP and port of the SNMP trap receiver in the SNMP Notifier using the following procedure:

1. Run the following command to edit the deployment:

```
kubectl edit deploy <snmp_notifier_deployment_name> -n <namespace>
```

Example:

```
kubectl edit deploy occne-snmp-notifier -n occne-infra
```

SNMP deployment yaml file is displayed.

2. Edit the SNMP destination in the deployment yaml file as follows:

```
--snmp.destination=<destination_ip>:<destination_port>
```


Example:

```
--snmp.destination=10.75.203.94:162
```

3. Save the file.

Checking SNMP Traps

Following is an example on how to capture the logs of the trap receiver server to view the generated SNMP traps:

```
docker logs <trapd_container_id>
```

Sample output:

```
NET-SNMP version 5.8 2024-04-19 08:32:01 <UNKNOWN> [UDP:
[10.121.27.121]:29375->[192.168.200.191]:162]: DISMAN-EVENT-
MIB::sysUpTimeInstance = Timeticks: (483032200) 55 days, 21:45:22.00 SNMPv2-
MIB::snmpTrapOID.0 = OID: ORACLENEF-MIB::OcnefQOSServiceDown ORACLENEF-
MIB::OcnefQOSServiceDown.1 = STRING:
"1.3.6.1.4.1.323.5.3.39.1.2.7021[job=occne-infra/occne-nf-cnc-podmonitor]"
ORACLENEF-MIB::OcnefQOSServiceDown.2 = STRING: "critical" ORACLENEF-
MIB::OcnefQOSServiceDown.3 = STRING: "Status: critical - Alert:
OcnefQOSServiceDown Summary: namespace: sanity, podname: , timestamp:
2024-04-19 08:32:01.354 +0000 UTC : QOS service down Description: OCNEF QOS
service qualityofservice is down" 2024-04-19 08:37:01 <UNKNOWN> [UDP:
[10.121.27.121]:6699->[192.168.200.191]:162]: DISMAN-EVENT-
MIB::sysUpTimeInstance = Timeticks: (483062200) 55 days, 21:50:22.00 SNMPv2-
MIB::snmpTrapOID.0 = OID: ORACLENEF-MIB::OcnefQOSServiceDown ORACLENEF-
MIB::OcnefQOSServiceDown.1 = STRING:
"1.3.6.1.4.1.323.5.3.39.1.2.7021[job=occne-infra/occne-nf-cnc-podmonitor]"
ORACLENEF-MIB::OcnefQOSServiceDown.2 = STRING: "clear" ORACLENEF-
MIB::OcnefQOSServiceDown.3 = STRING: "Status: OK"
```

MIB Files for NEF

There are two MIB files which are used to generate the traps. The user need to update these files along with the Alert file in order to fetch the traps in their environment.

- `ocnef_mib_tc_24.2.0.mib`: This is considered as NEF and CAPIF top level mib file, where the Objects and their data types are defined.
- `ocnef_mib_24.2.0.mib`: This file fetches the Objects from the top level mib file and based on the Alert notification for NEF, these objects can be selected for display.
- `ocnef_capif_mib.mib`: This file fetches the Objects from the top level mib file and based on the Alert notification for CAPIF, these objects can be selected for display.

Note

MIB files are packaged with `Custom_Templates.zip`. You can download the file from [My Oracle Support](#) as described in Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide.

7.3 NEF Alert Configuration in OCI

The following procedure describes how to configure the NEF alerts for OCI. The OCI supports metric expressions written in MQL (Metric Query Language) and thus, requires a new NEF alert file for configuring alerts in OCI observability platform.

The following are the steps:

1. Run the following command to extract the .zip file:

```
unzip ocnef_oci_alertrules_<version>.zip
```

The `occapif_oci_alarm_terraform` and `ocnef_oci_alarm_terraform` folders are available in the zip file.

Note

The zip file is available in the `custom_templates.zip`.

2. Open the `occapif_oci_alarm_terraform` folder, in the `notifications.tf` file, update the parameter endpoint with the email id of the user.
3. Open the `ocnef_oci_alarm_terraform` folder, in the `notifications.tf` file, update the parameter endpoint with the email id of the user (replace `test@gmail.com` with the email id of the user).
4. Log in to the OCI Console.

Note

For more details about logging in to the OCI, refer to [Signing In to the OCI Console](#).

5. Open the navigation menu and select **Developer Services**. The **Developer Services** window appears in the right pane.
6. Under the **Developer Services**, select **Resource Manager**.
7. Under **Resource Manager**, select **Stacks**. The **Stacks** window appears.
8. Click **Create Stack**.
9. Select the default **My Configuration** radio button.
10. Under Stack configuration, select the folder radio button and upload the `occapif_oci_alarm_terraform` folder.
11. Enter the **Name** and **Description** and select the **compartment**.
12. Select the latest Terraform version from the **Terraform version** drop-down.
13. Click **Next**. The **Edit Stack** screen appears.
14. Enter the required inputs to create the NEF alerts or alarms and click **Save** and **Run Apply**.
15. Verify that the alarms are created in the Alarm Definitions screen (**OCI Console > Observability & Management > Monitoring > Alarm Definitions**) provided.

The required inputs are:

- **Alarms Configuration**
 - **Compartment Name** - Choose name of compartment from the drop-down
 - **Metric namespace** - Metric namespace that the user provided while deploying OCI Adaptors.
 - **Topic Name** - Any user configurable name. Must contain fewer than 256 characters. Only alphanumeric characters plus hyphens (-) and underscores (_) are allowed.
 - **Message Format** - Keep it as ONS_OPTIMIZED. (This is pre-populated)
 - **Alarm is_enabled** - Keep it as **True**. (This is pre-populated)
- 16. The steps 6 to 16 must be repeated for uploading the `ocnef_oci_alarm_terraform` folder. Keep **Metric namespace** as **mgmtagent_kubernetes_metrics** (This is pre-populated).

7.3.1 Configuring NEF Alerts for OCI

To configure NEF alerts for OCI, OCI supports metric expressions written in MQL (Metric Query Language) and therefore requires `ocnef_oci_alertrules_24.2.2.zip` file for configuring alerts in OCI observability platform. For more information, see *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*.