Oracle® Communications

Network Repository Function (NRF) Cloud Native User's Guide

Release 1.10.0
F37437-01
January 2021

ORACLE

Oracle Communications Network Repository Function (NRF) Cloud Native User's Guide, Release 1.10.0

F37437-01

Contents

10 OCNRF Metrics, KPIs, and Alerts

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:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
    - 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**.

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.

What's New in This Guide

This section introduces the documentation updates for Release 1.10.x in Oracle Communications Cloud Native Network Repository Function (NRF) User's Guide.

New and Updated Features in Release 1.10.0

- Added new attributes to NF Discovery Options to enable discover validity period per NFType and updated NF Discovery Options CNC Console section for discovery validity period configuration.
- Added Rest based OCNRF state data retrieval feature section.
- Added new attributes related to Key ID in NF Access Token Options section.
- Added `useOAuthToken` attribute in SLF Options section.
- New metrics dimensions and metrics are added in OCNRF Metrics section.
- OCNRF supports for NFSetId parameter of 3GPP 29510 Release 16.3.0 in the following service operations:
  - NfRegister
  - NfUpdate
  - NfSubscription
  - NfDiscover
  - NfAccessToken
- Support for SCP Registering with NRF as 3GPP 29510 Release 16.3.0 NFType: OCNRF now allows SCP to register as a 3gpp defined NF Type. It also allows Consumer NFs to subscribe, discover and request Access Token for SCP as 3gpp defined NF Type.

1 Introduction

This document provides information about the role of Oracle Communications Network Repository Function (OCNRF) in 5G Service Based Architecture and how to configure and use OCNRF.

OCNRF is a key component of the 5G Service Based Architecture. OCNRF maintains an updated repository of all the Network Functions (NFs) available in the operator's network along with the services provided by each of the NFs in the 5G core that are expected to be instantiated, scaled and terminated with minimal to no manual intervention. In addition to serving as a repository of the services, OCNRF also supports discovery mechanisms that allows NFs to discover each other and get updated status of the desired NFs.

OCNRF supports the following functions:

- Maintains the profiles of the available NF instances and their supported services in the 5G core network.
- Allows consumer NF instances to discover other providers NF instances in the 5G core network.
- Allows NF instances to track the status of other NF instances.
- Provides Oauth2 based Access Token service for consumer NF authorization.
- Provides specific NF Type selection based on subscriber identity.
- Supports forwarding of messages from one NRF to another NRF.
- Supports geo-redundancy to ensure service availability.

The OCNRF interacts with every other NF in the 5G core network and it supports the above functions through the following services:

- Management Services
- Discovery Services
- AccessToken Service

Acronyms

The following table provides information about the acronyms and the terminology used in the document.

Table 1-1 Acronyms

Term	Definition
3GPP	3rd Generation Partnership Project
5G-AN	5G Access Network
5GC	5G Core Network

Table 1-1 (Cont.) Acronyms

Term	Definition
5G System	3GPP system consisting of 5G Access Network (AN), 5G Core Network and UE
AMF	Access and Mobility Management Function
API Gateway	Application that sits in front of an application programming interface (API) and acts as a single point of entry for a defined group of micro services.
CNE	Cloud Native Environment
Dimension	Dimension is a tag of Metric. For Example, "ocnrf_nfRegister_rx_requests_total {{ OriginatorNfType }} {{NrfLevel }} {{NfInstanceId }}" In the example above, OriginatorNfType, NrfLevel, and NfInstanceId are dimensions.
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
KBs	Kubernetes
KPI	Key Performance Indicator
MMI	Machine Machine Interface
MPS	Messages Per Second
NDB	Network Database
NF	Network Function
Network Function	A functional building block within a network infrastructure, which has well defined external interfaces and well defined functional behavior. In practical terms, a network function is often a network node or physical appliance.
Network Slice	A logical network that provides specific network capabilities and network characteristics.
Network Slice instance	A set of Network Function instances and the required resources (For Example, compute, storage and networking resources) which form a deployed Network Slice.
NF Consumer	A generic way to refer to an NF which consumes services provided by another NF. For Example: An AMF is referred to as a Consumer when it consumes AMPolicy services provided by the PCF.
NF Instance	A specific instance of a network function type.
NF Producer or NF Provider	A generic way to refer to an NF which provides services that can be consumed by another NF. For Example: A PCF is a provider NF and provides AMPolicy Services
NRF	Network Repository Function or Network Function Repository Function
PCF	Policy Control Function
PLMN	Public Land Mobile Network
Resiliency	The ability of the NFV framework to limit disruption and return to normal or at a minimum acceptable service delivery level in the fame of a fault, failure, or an event that disrupts normal operation.

Table 1-1 (Cont.) Acronyms

Term	Definition
Scaling	Ability to dynamically extend/reduce resources granted to the Virtual Network Function (VNF) as needed. This includes scaling out/in or scaling up/down.
Scaling Out/In/ Horizontally	The ability to scale by add/remove resource instances (For Example, VMs). Also called scaling Horizontally.
Scaling Up/Down/ Vertically	The ability to scale by changing allocated resources, for example increase/decrease memory, CPU capacity or storage size.
SCP	Service Communication Proxy
SEPP	Security Edge Protection Proxy
SLF	Subscriber Location Function
SMF	Session Management Function
URI	Uniform Resource Identifier

OCNRF References

- Cloud Native Environment 1.6 Installation Guide
- OCNRF Installation and Upgrade Guide
- CNC Console User's Guide
- ATS User Manual

2 OCNRF Supported Services

This section includes information about the services supported by OCNRF.

OCNRF supports the following services:

OCNRF Management Services

The OCNRF Management service is identified by the service operation name `Nnrf_NFManagement`.

OCNRF supports the following management services:

> **Note:**
>
> The respective service operation name is mentioned next to each service.

- **Register NF instance** (`NFRegister`): Allows the NF instance to register its NF profile in the OCNRF along with the list of services provided by the NF instance.
- **Update NF instance** (`NFUpdate`): Enables the NF instance to partially update or replace the parameters of its NF profile in the OCNRF. It also allows to add or delete services provided by the NF instance.
  This operation supports the following:
  - Complete replacement of NF profile
  - Add, remove, or update attributes of NF Profile
  - Heart beat and load information of NF
- **De-register NF instance** (`NFDeregister`): Enables the NF instance to de-register its NF profile and the services provided by the NF instance from the 5G network.
- **Subscribe to Status** (`NFStatusSubscribe`): Enables the NF instance to subscribe the status changes of other NF instances registered in the OCNRF.
- **Unsubscribe to Status** (`NFStatusUnsubscribe`): Enables the NF instance to unsubscribe the status changes of other NF instances.
- **Notifications of Status** (`NFStatusNotify`): Sends status notifications to subscribed NFs.
- **Retrieval of NF list** (`NFListRetrieval`): Allows the retrieval of a list of NF Instances that are currently registered in OCNRF. This service operation is not allowed to be invoked from the OCNRF in a different PLMN.
- **Retrieval of a NF Profiles** (`NFProfileRetrieval`): Allows the retrieval of the NF profile of a given NF instance currently registered in OCNRF. This service operation is not allowed to be invoked from the OCNRF in a different PLMN.

OCNRF Discovery Service

The OCNRF Discovery service is identified by the service operation name `Nnrf_NFDiscovery Service`.

OCNRF supports the following discovery service:

- **Discover NF instance** (`NFDiscover`): OCNRF supports discovery of OCNRF Profile of the NF instances, or NF Services that match certain input criteria.

OCNRF Access Token Service

The OCNRF Access Token service handles 3GPP defined AccessToken service operations. Oauth2.0 based token is provided by OCNRF according to inputs provided by consumer network function in access token request.

OCNRF supports the following access token service:

- Access Token (`Nnrf_AccessToken`): OCNRF supports issuing OAuth2 token to consumer NFs for accessing specific Producer Services.

3
OCNRF Architecture

OCNRF comprises of various microservices deployed in Kubernetes based Cloud Native Environment (CNE, example: OCCNE). Some common services like logs or metrics data collection, analysis and graphs or charts visualization, and so on are provided by the environment. The microservices integrate with the environment and provide the necessary data.

Following are the components of OCNRF product:

- **NF Registration Microservice**
  This microservice receives and handles the following service operations:
  - NFRegister service requests from the NFs
  - NFUpdate service requests from the NFs
  - NFDeregister service requests from the NFs
  - NFListRetrieval service requests from the NFs
  - NFProfileRetrieval service requests from the NFs
  - Heart-beat messages from the NFs
- **NF Subscription Microservice**
  This microservice performs the following service operations:
  - receives and handles NFStatusSubscribe service requests from the NFs
  - receives and handles NFStatusUnsubscribe service requests from the NFs
  - sends NFStatusNotify service requests towards the subscribed NFs
- **NF Discover Microservice**
  This microservice receives and handles the following service operations:
  - NFDiscover service requests from the NFs
- **NF AccessToken Microservice**
  This microservice handles 3GPP defined AccessToken service operations. Oauth2.0 based token is provided by OCNRF according to inputs provided by consumer network function in access token request.
- **OCNRF Auditor Microservice**
  This microservice is internal to OCNRF. This microservice performs the following tasks:
  - finds and deletes the expired subscription records
  - finds and deletes the profile records which have been SUSPENDED for a very long time
  - monitors the heart-beat expiry, mark the NF profiles as suspended and act appropriately on the suspended NF profiles
- **OCNRF Configuration Microservice**
  This microservice is used to configure OCNRF. These configuration can be changed dynamically by a operator/user using REST based interface. This

configuration data is managed by the OCNRF configuration service and is stored in a separate data store.

- **OCNRF Ingress Gateway Microservice**
  This microservice is entry point for accessing OCNRF supported service operations.
- **OCNRF Egress Gateway Microservice**
  This microservice is responsible to route OCNRF initiated egress messages to other NFs.
- **App Info Microservice**
  This microservice is responsible to get the status of microservices related to NFManagement Service operations (that is NF Registration microservice, NF Subscription microservice, NRF Auditor microservice).

  NF Registration, NF Subscription, NRF Auditor microservices uses this microservice to get the status of other two microservices. In case any of service is found as down, status asking microservice will reject the incoming messages. This microservice is also responsible to fetch DB replication status whether it is active or not-active.

4
OCNRF Features

This section explains the OCNRF features.

Subscriber Location Function

OCNRF supports Subscriber Location Function (SLF) feature which identifies specific NF Producer selection based on Subscriber Identity (like SUPI and GPSI) as explained below:

- Checks if SLF lookup needs to be performed before processing the discovery request
  - If GroupId is already present in the Discovery Query, even if SLF feature is ENABLED. OCNRF will use the received Group Id and will not perform SLF lookup.
  - Supported NF Types for which SLF lookup will be performed can be configured using "supportedNfTypeList" attributes in SLF Options. OCNRF supports the below NF Type(s) for performing SLF lookup:
    * UDM
    * UDR
    * AUSF
- Finds the NF Group Id by sending SLF query (that is, Nudr_GroupIDmap service operation) with the received Subscriber Identity (like SUPI and GPSI).
- Generates NFDiscover service response using the NF Group Id received in SLF response and other parameters.
  - The received Subscriber Identifier will not be used during NF Producer selection.

Key points:

- SLF function on UDR network function will be configured with details of subscribers mapped to group identifiers. Producer network functions will be deployed in network belonging to group identifiers
- Producer network functions will not publish the range of subscriber identities in their network profile while registering with OCNRF
- While discovering the producer network functions, consumer Network functions will send discover query parameters including the subscriber identities like SUPI, GPSI. Note mandatory attributes of NFDiscover service operation will be in discovery query as per 3GPP specification 29.510 v 15.5.0. Consumer may provide additional query parameters provided in 3GPP specification 29.510 v 15.5.0
- OCNRF will need to send query to UDR with subscriber identity received in NF Discover service operation
- UDR will provide group id mapped to subscriber identity

- OCNRF will do discovery of network function using the group id provided by UDR/SLF and send response back to consumer network function with discovered producer network function profiles

Refer to SLF Options for more information on REST APIs.

For metrics, alerts and KPIs, refer to NRF-SLF Metrics , OCNRF Alerts and OCNRF KPIs sections respectively.

See SLF Options to know how to configure SLF using CNC Console.

OCNRF Forwarding

OCNRF Forwarding feature forwards the service operation messages if OCNRF is not able to fulfill the required service operation.

> **Note:**
>
> Service operations with specific cases or scenarios are eligible for forwarding.

A consumer NF instance can perform the following:

- Subscribe to changes of NF instances registered in an NRF to which it is not directly interacting. The NF subscription message is forwarded by an intermediate NRF to another NRF.
- Retrieve the NF profile of the NF instances registered in an NRF to which it is not directly interacting. The NF profile retrieval message is forwarded by an intermediate NRF to another NRF.
- Discover the NF profile of the NF instances registered in an NRF to which it is not directly interacting. The NF discover message is forwarded by an intermediate NRF to another NRF.
- Request OAuth 2.0 access token for the NF instances registered in an NRF to which it is not directly interacting. The OAuth 2.0 access token service request is forwarded by an intermediate NRF to NRF (which may issue the token).

Refer to NRF-NRF Forwarding Options for more information on REST APIs.

For metrics, alerts and KPIs, refer to NRF Forwarding Metrics, OCNRF Alerts and OCNRF KPIs sections respectively.

See Forwarding Options to know how to configure OCNRF forwarding feature using CNC Console.

OCNRF Geo-Redundancy

OCNRF supports two site Geo-Redundancy to ensure service availability when one OCNRF site is down. When OCNRF is deployed as Geo-Redundant NRF, both the OCNRFs works in Active state. The NFs in a given site needs to configure one of the Geo-Redundant OCNRF as the primary NRF and the other one as secondary NRF. If the primary OCNRF is available, the NFs shall send service requests to the primary OCNRF. In case the primary OCNRF is down, the NF shall redirect its traffic to the secondary OCNRF till the primary OCNRF is unavailable.

The OCNRF's State data gets replicated between the Geo-Redundant sites by using DB tier's replication service.

With OCNRF Geo-Redundant feature, availability of OCNRF's services will work as below:

- Unavailability of any one of NFRegistration, NFSubscription, and NrfAuditor micro-services will result in unavailability of NFManagement service operations at OCNRF.
- NFDiscovery and NFAccessToken services of OCNRF will continue to work as independent service operation.

Following are the requirements for geo-redundancy:

- Both geo-redundant sites must have helm and rest based configuration (except NRF Instanced Id, OCNRF host and port).
- Geo-Redundant sites must be time synchronized.
- Geo-Redundant OCNRF sites must be reachable from NFs on both sites.
- NFs needs to configure Geo-Redundant OCNRF details as Primary and Secondary NRFs.
- NFs must not communicate to both Geo-Redundant OCNRF sites at same time.

Refer to Geo Redundancy Options for more information on REST APIs.

For metrics, alerts and KPIs, refer to GeoRedundancy metrics , OCNRF Alerts and OCNRF KPIs sections respectively.

See Geo Redundancy Options to know how to configure OCNRF geo-redundancy feature using CNC Console.

NF Heartbeat enhancement

This feature allows the operator to configure minimum, maximum, default heartbeat Timers and the maximum number of consecutive heartbeats the NF is allowed to miss. Further, these values can customized per NF type.

According to 3GPP 29.510, every NF registered with the NRF keeps its operative status alive by sending NF heartbeat requests periodically. The NF can optionally send the heartbeatTimer value when it registers its NFProfile or when it wants to update its registered NFProfile.

OCNRF may modify the value of the heartbeatTimer based on its configuration and return the new value to the NF on successful registration. The NF will thereafter use the heartbeatTimer as received in the registration response as its heartbeat interval.

In case the configuration changes at the OCNRF for the heartbeatTimer, the changed value must be communicated to the NF in the response of the next periodic NF Heart-Beat request or when it next sends a NFUpdate request to the OCNRF.

OCNRF monitors the operative status of all the NFs registered with it, and when it detects that an NF has missed updating its NFProfile or sending a heartbeat for the heartbeat interval, NRF must mark the NFProfile as SUSPENDED. The NFProfile and its services may not longer be discoverable by the other NFs through the NfDiscovery Service. Further, the NRF will notify the subscribed NFs of the change of status of the NFProfile.

OCNRF NF Authentication using TLS certificate

HTTPS support is a minimum requirement for 5G NFs as defined in 3GPP TS 33.501. This feature enables extending identity validation from Transport layer to the Application layer and also provides a mechanism to validate the NF FQDN presence in TLS certificate as added by the Service Mesh against the NF Profile FQDN present in the request.

This feature is used by OCNRF to authenticate the Network Function before accessing the OCNRF services. In case, authentication fails, service operation request will be rejected. In this feature, some attributes from TLS certificate is challenged against defined attributes.

OCNRF provides configurations to enable/disable the feature dynamically. To enable the feature on Ingress API-GW in OCNRF deployment. Refer to `xfccHeaderValidation` attribute in User Configurable Section of *OCNRF Installation Guide* for more details.

Refer to NF Authentication Options for more information on REST APIs.

For metrics, alerts and KPIs, refer to NF Authentication Metrics, OCNRF Alerts and OCNRF KPIs sections respectively.

See NF Authentication Options to know how to configure OCNRF authentication feature using CNC Console.

Access Token Request Authorization

NRF follows and supports the 3GPP 29.510 based verification for Access Token Authorization requests for specific NF-Producer based on the allowed NF Type, PLMN present in the NFProfiles. Extension to this requirement is to include screening for Access Token requests based on NF Type.

NRF plays major role as an OAuth2.0 Authorization server in 5G Service based architecture. When a NF service Consumer needs to access the services of a NF producer of a particular NFType and NFInstanceId, it obtains an OAuth2 access token from the NRF. NRF performs the required authorization, and if successful, will issue the token with the requested claims. Using this feature, OCNRF provides an option to the user to tailor the authorization of the Producer-Consumer NF Types along with the Producer NF's services.

User can configure mapping of the Requester NF Type, Target NF Type, and the allowed services of the Target NF. Access Token request received based on the configuration and is furthered processes the request only if the authorization is successful. Allowed Services can be configured as single wild card '*' which denotes all the TargetNfs services are allowed for the consumer NF. User can also configure the HTTP status code and error description that will be used in the Error Response sent by the OCNRF when an Access Token request is rejected.

Access Token configurable attribute "`logicalOperatorForScope`" is used while authorizing the services in the Access Token Request's scope against the allowed services in the configuration. If the `logicalOperatorForScope` is set to "OR", at-least one of the services in the scope will be present in the allowed services. If it is set to "AND", all the services in the scope will be present in the allowed services.

Refer to NF Access Token Options for more information on REST APIs.

For metrics, alerts and KPIs, refer to NF AccessToken Authorization Metrics, OCNRF Alerts and OCNRF KPIs sections respectively.

See NF Access Token Options to know how to configure OCNRF AccessToken feature using CNC Console.

Service mesh for intra-NF communication

Oracle NRF leverages the Istio or Envoy service mesh (Aspen Service Mesh) for all internal and external communication. The service mesh integration provides inter-NF communication and allows API gateway co-working with service mesh. The service mesh integration supports the services by deploying a special sidecar proxy in the environment to intercept all network communication between microservices. Refer to *OCNRF Installation guide* for more details on configuring ASM.

NF Screening

NF Screening supports the functionality to screen the service requests received from 5G Network Functions (NFs) before allowing access to OCNRF services.

In this feature, OCNRF screens the incoming service operations from NFs on the basis of some attributes against set of rules configured at OCNRF. OCNRF processes the required services only if screening is successful.

This feature provides extra security by restricting the NF that can use the service of OCNRF. Operator can decide which NF with required attributes can access the services provided by OCNRF. To implement this, operator can configure various screening lists in which attributes can be configured to tell which attribute is allowed or not.

> **Note:**
>
> By default, NF Screening feature is globally disabled. This feature can be enabled by setting the **nfScreeningRulesListStatus** attribute as "ENABLED" using REST based Interface.

The screening can be in the form of **Whitelist** or **Blacklist**.

- When a screening list is configured to operate as a whitelist, the request is allowed to access the service only if the corresponding attribute value is present in the whitelist.
- When a screening list is configured to operate as a blacklist, the request is allowed to access the service only if the corresponding attribute value is not present in the blacklist.

Screening Lists can have rules for ***global*** and per NF type:

- The global level screening lists allows operators to configure screening that is common to all NFs.
- Per NF Type level rules provides additional flexibility/granularity for screening that can be controlled on a per NF type basis.

> **Note:**
>
> - The rules can be configured at both Global level and Per NF Type level.
> - "NF type list allowed to Register" is available at Global level only.

Refer to NF Screening Configuration for more information on REST APIs.

For metrics, alerts and KPIs, refer to NF Screening Metrics, OCNRF Alerts and OCNRF KPIs sections respectively.

See NF Screening Options to know how to configure OCNRF AccessToken feature using CNC Console.

Rest based OCNRF state data retrieval feature

Rest based OCNRF state data retrieval feature provide options for Non-Signaling APIs to access OCNRF state data. It helps operator/user to access the OCNRF state data to understand and debug in the event of failures.

This feature provides various queries which can help to get data as per the need. Different query attributes helps to achieve the required data.

Refer to OCNRF state data retrieval REST APIs for more information on REST APIs.

KeyID for Access Token

Key-ID feature is about to add "kid" header in Access Token Response generated by OCNRF. As per RFC 7515 section 4.1.4, the use of Key-ID (KID) is to hint indicate which key was used to secure JWS.

Each OCNRF and NF producer(s) will have multiple keys with algorithm indexed with "kid" configured. OCNRF Rest based configuration provides options to configure different key-ids and corresponding OCNRF private keys and public certificates along with corresponding oauth signing algorithms. One of the configured key-ids, can be set as current key-id.While generating the oauth access token, OCNRF uses the keys, algorithm and certificates corresponding to current key-id.

OCNRF configuration provides "addkeyIDInAccessToken" attribute which tells to add key-id as header in Access Token Response or not. If value is true, then currentKeyID value will be added in "kid" header in AccessToken Response. If value is false, then "kid" header will not be added in AccessToken Response.

Refer to NF Access Token Options for more information on REST API configuration.

Refer to OCNRF configuration status REST APIs for more information on how to check AccessToken Signing Key Status.

Refer to OCNRF Alerts and OCNRF KPIs sections for more information on alerts and KPIs.

Automated Test Suite Support

OCNRF provides Automated Test Suite for validating the functionalities. ATS allows you to execute OCNRF 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. Refer to ATS User Manual for more information.

5
Configuring OCNRF

OCNRF can be configured using HELM and REST configuration. Some configuration are performed during installation using HELM and few are modified using REST. For HELM configuration refer to *OCNRF Cloud Native Installation and Upgrade Guide*. The REST configurations can also be performed using Cloud Native Core (CNC) Console. Refer to Configuring OCNRF using CNC Console for more details.

Mandatory Configurations

Following are the mandatory parameter, which must be configured before using OCNRF:

- nrfPlmnList: PLMN(s) served by OCNRF. This must be configured before using any OCNRF Services.
- ocnrfHost: OCNRF Host's FQDN.
- ocnrfPort: OCNRF Host's Port.

Subscriber Location Function Configuration

SLF Host Configuration attribute (`slfHostConfig`) allows the user to configure the details of SLF/UDR Network Functions.

The `slfHostConfig` configuration consists of attributes like apiVersion, scheme, FQDN, port, priority, etc. OCNRF allows to configure more than two host details. However the host with highest priority is considered as Primary Host. The host with second highest priority is considered as Secondary Host.

> **Note:**
>
> - Refer 29.510, release 15.5 for definition and allowed range for `slfHostConfig` attributes (apiVersion, scheme, FQDN, port, priority, etc).
> - Apart from priority attribute, no other attributes plays any role in Primary/ Secondary host selection.
> - Apart from Primary/Secondary host, other configured hosts (if any) are not used during any message processing.
> - When more than one host is configured with highest priority, then two of them will be picked as Primary/Secondary host randomly.

In Subscriber Location Function (SLF) feature, SLF request is first sent to Primary SLF. In case of error from Primary SLF, request is sent to Secondary SLF based on below configuration:

1. rerouteOnResponseHttpStatusCodes: This configuration is used to determine if the SLF request message can be sent to Secondary SLF or not. After getting response from primary SLF, if response status code from primary SLF matches with this configuration, then OCNRF reroutes the request to the secondary SLF. Refer `slfHostConfig` attribute for Primary and Secondary SLF details.
2. maximumHopCount: This configuration is used to determine Maximum number of hops (SLF/NRF) that OCNRF can forward a given service request. This configuration more useful during NRF Forwarding and SLF feature interaction.

NRF Forwarding

NRF Host Configuration attribute (`nrfHostConfig`) allows the user to configure the details of another NRF to which forwarding will happen.

> **Note:**
>
> For NRF-NRF forwarding feature to work, `nrfHostConfig` attribute must be configured in both NRFs with each other values.
> For example: OCNRF1 is forwarding requests towards OCNRF2, then `nrfHostConfig` configuration attribute of OCNRF1 shall have OCNRF2 details and similarly `nrfHostConfig` configuration attribute of OCNRF2 shall have OCNRF1 details. These configurations are used while handling different service operations of OCNRF.

The `nrfHostConfig` configuration consists of attributes like apiVersion, scheme, FQDN, port, priority, etc. OCNRF allows to configure more than two host details. However the host with highest priority is considered as Primary Host. The host with second highest priority is considered as Secondary Host.

> **Note:**
>
> - Refer 29.510, release 15.5 for definition and allowed range for NfHostConfig attributes (apiVersion, scheme, FQDN, port, priority, etc).
> - Apart from priority attribute, no other attributes plays any role in Primary/Secondary host selection.
> - Apart from Primary/Secondary host, other configured hosts (if any) are not used during any message processing.
> - When more than one host is configured with highest priority, then two of them will be picked as Primary/Secondary host randomly.

In NRF forwarding feature, request is first forwarded to Primary NRF. In case of error, request is forwarded to Secondary NRF based on below configuration:

1. nrfRerouteOnResponseHttpStatusCodes: This configuration is used to determine if the service operation message can be forwarded to Secondary NRF or not. After getting response from primary NRF, if response status code from primary NRF matches with this configuration, then OCNRF reroutes the request to the

secondary NRF. Refer `nrfHostConfig` attribute for Primary and Secondary NRF details.

2. maximumHopCount: This configuration is used to determine Maximum number of hops (SLF/NRF) that OCNRF can forward a given service request. This configuration more useful during NRF Forwarding and SLF feature interaction.

Configuring Access Token Request Authorization

Configuration for OCNRF Access Token Request Authorization

With `nfAccessTokenOptions` API, `authFeatureConfig` attribute provides the support required to use OCNRF Access Token Request Authorization Feature. Refer to OCNRF Configuration REST APIs for more details.

Sample configuration to use the feature

```
{
   "oauthTokenExpiryTime": "1h",
   "authorizeRequesterNf": "ENABLED",
   "logicalOperatorForScope": "AND",
   "audienceType": "NF_INSTANCE_ID",
   "authFeatureConfig": {
      "featureStatus": "DISABLED",
      "authRulesConfig": [
            {
               "targetNfType":"PCF",
               "requesterNfType":"AMF",
               "serviceNames":[
                  "npcf-am-policy-control",
                  "npcf-eventexposure"
               ]
            },
            {
               "targetNfType":"UDM",
               "requesterNfType":"AMF",
               "serviceNames":[
                  "*"
               ]
            }
         ],
      "errorResponses": [{
         "errorCondition": "RequesterNf_Unauthorized",
         "responseCode": 400,
         "errorResponse": "The Consumer NfType is not authorized to
receive access token for the requested Nftype.",
         "retryAfter": "5m",
         "redirectUrl": ""
     }]
   },
   "tokenSigningDetails": {
        "currentKeyID": "a14ef8e1bc5c",
        "addkeyIDInAccessToken": true,
        "defaultK8SecretDetails": {
            "k8SecretName": "ocnrf",
```

```
            "k8SecretNameSpace": "ocnrf"
        },
        "keyDetailsList": [
            {
                "keyID": "a14ef8e1bc5c",
                "algorithm": "ES256",
                "privateKey": {
                    "k8SecretName": "ocnrf",
                    "k8SecretNameSpace": "ocnrf",
                    "fileName": "ec_private_key_pkcs8.pem"
                },
                "certificate": {
                    "k8SecretName": "ocnrf",
                    "k8SecretNameSpace": "ocnrf",
                    "fileName": "ecdsa_ocnrfapigatewayTestCA.cer"
                }
            }
        ]
    },
   "errorResponses": [
        {
            "errorCondition": "Invalid_Key_Details",
            "responseCode": 500,
            "errorResponse": "Configured Key ID details are invalid and
cannot be used",
            "retryAfter": "5m",
            "redirectUrl": ""
        },
        {
            "errorCondition": "Current_Key_Id_Not_Configured",
            "responseCode": 500,
            "errorResponse": "Current Key ID is not configured",
            "retryAfter": "5m",
            "redirectUrl": ""
        }
    ]
}
```

Configuring NF Authentication using TLS certificate

This feature is used by OCNRF to authenticate the Network Function before accessing the OCNRF services. In case, authentication fails, service operation request is rejected. In this feature, some attributes from TLS certificate is challenged against defined attributes.

OCNRF provides configuration to enable/disable the feature dynamically. Refer to `xfccHeaderValidation` attribute in User Configurable Section of *OCNRF Installation Guide* to enable the feature on Ingress API gateway in OCNRF deployment.

Note:

- This feature is disabled by default. Feature needs to be enabled at API-GW and OCNRF levels both to make this feature work. At OCNRF level, feature enabling/disabling can be done using mentioned configuration below.
- Once this feature is enabled. All of NFs must re-register with FQDN in NF Profile or NFs can send NFUpdate with FQDN. For Subscription Service Operations, Network Functions need to register with OCNRF, even NFs has taken Subscription Prior to enabling the feature, need to Register with NRF for further service operations.

Configuration Required to use OCNRF NF Authentication using TLS certificate feature

Refer to attributes under `nfAuthenticationSystemOptions` in NF Authentication Options for more details.

Sample configuration to use the feature

Enable the following attributes:

```
{

    "nfRegistrationStatus": "DISABLED",
    "nfSubscriptionStatus": "DISABLED",
    "nfDiscoveryStatus": "DISABLED",
    "accessTokenStatus": "DISABLED",
    "nfProfileStatus": "DISABLED",
    "nfListRetrievalStatus": "DISABLED",
    "checkIfNfIsRegistered": "DISABLED",
    "errorResponses": [{
    "errorCondition": "Nf_Fqdn_Authentication_Failure",
       "responseCode": 403,
       "errorResponse": "Failed to authenticate NF using FQDN",
       "retryAfter": "5m",
       "redirectUrl": ""
   }]
}
```

6
OCNRF Configuration REST APIs

Service API Interfaces

This section lists the API interface details for each OCNRF services.

Table 6-1 Service API Interfaces

Resource Name	Resource URI	HTTP Method	Data Model for Request	Data Model for Response	Description
ocnrfConfigurations	{apiRoot}/nrf-configuration/v1/	GET	Not applicable	ocnrfConfigurations	Retrieve all of the OCNRF configurations in single GET request. This includes all Options and NFScreeningRules.
generalOptions	{apiRoot}/nrf-configuration/v1/generalOptions	GET	Not applicable	generalOptions	Retrieve OCNRF general options configuration.
generalOptions	{apiRoot}/nrf-configuration/v1/generalOptions	PUT	generalOptions	generalOptions	Updates OCNRF general options configuration.
nfScreeningOptions	{apiRoot}/nrf-configuration/v1/nfScreeningOptions	GET	Not applicable	nfScreeningOptions	Retrieve OCNRF NF Screening options configuration.
nfScreeningOptions	{apiRoot}/nrf-configuration/v1/nfScreeningOptions	PUT	nfScreeningOptions	nfScreeningOptions	Updates OCNRF NF Screening options configuration.
nfManagementOptions	{apiRoot}/nrf-configuration/v1/nfManagementOptions	GET	Not applicable	nfManagementOptions	Retrieve OCNRF NF Management options configuration.
nfManagementOptions	{apiRoot}/nrf-configuration/v1/nfManagementOptions	PUT	nfManagementOptions	nfManagementOptions	Updates OCNRF NF Management options configuration.
nfDiscoveryOptions	{apiRoot}/nrf-configuration/v1/nfDiscoveryOptions	GET	Not applicable	nfDiscoveryOptions	Retrieve OCNRF NF Discovery options configuration.
nfDiscoveryOptions	{apiRoot}/nrf-configuration/v1/nfDiscoveryOptions	PUT	nfDiscoveryOptions	nfDiscoveryOptions	Updates OCNRF NF Discovery options configuration.
nfAccessTokenOptions	{apiRoot}/nrf-configuration/v1/nfAccessTokenOptions	GET	Not applicable	nfAccessTokenOptions	Retrieve OCNRF NF AccessToken options configuration.
nfAccessTokenOptions	{apiRoot}/nrf-configuration/v1/nfAccessTokenOptions	PUT	nfAccessTokenOptions	nfAccessTokenOptions	Updates OCNRF NF AccessToken options configuration.

Table 6-1 (Cont.) Service API Interfaces

Resource Name	Resource URI	HTTP Method	Data Model for Request	Data Model for Response	Description
forwardingOptions	{apiRoot}/nrf-configuration/v1/forwardingOptions	GET	Not applicable	forwardingOptions	Retrieve OCNRF Forwarding options configuration.
forwardingOptions	{apiRoot}/nrf-configuration/v1/forwardingOptions	PUT	forwardingOptions	forwardingOptions	Updates OCNRF Forwarding options configuration.
slfOptions	{apiRoot}/nrf-configuration/v1/slfOptions	GET	Not applicable	slfOptions	Retrieve OCNRF SLF options configuration.
slfOptions	{apiRoot}/nrf-configuration/v1/slfOptions	PUT	slfOptions	slfOptions	Updates OCNRF SLF options configuration.
geoRedundancyOptions	{apiRoot}/nrf-configuration/v1/geoRedundancyOptions	GET	Not applicable	geoRedundancyOptions	Retrieve OCNRF Geo Redundancy options configuration.
geoRedundancyOptions	{apiRoot}/nrf-configuration/v1/geoRedundancyOptions	PUT	geoRedundancyOptions	geoRedundancyOptions	Updates OCNRF Geo Redundancy options configuration.
nfAuthenticationOptions	{apiRoot}/nrf-configuration/v1/nfAuthenticationOptions	GET	Not applicable	nfAuthenticationOptions	Retrieve OCNRF NF Authentication options configuration.
nfAuthenticationOptions	{apiRoot}/nrf-configuration/v1/nfAuthenticationOptions	PUT	nfAuthenticationOptions	nfAuthenticationOptions	Updates OCNRF NF Authentication options configuration.
logLevelOptions	{apiRoot}/nrf-configuration/v1/logLevelOptions	GET	Not applicable	logLevelOptions	Retrieve OCNRF Log Level options configuration.
logLevelOptions	{apiRoot}/nrf-configuration/v1/logLevelOptions	PUT	logLevelOptions	logLevelOptions	Updates OCNRF Log Level options configuration.
screening-rules	{apiRoot}/nrf-configuration/v1/screening-rules	GET	Not applicable	ScreeningRulesResult	Returns all the screening rules.
screening-rules	{apiRoot}/nrf-configuration/v1/screening-rules	GET	nfScreeningRulesListType or/and nfScreeningRulesListStatus	ScreeningRulesResult	Returns screening rules corresponding to the specified NF Screening Rule List Type. Query:- {apiRoot}/nrf-configuration/v1/screening-rules?nfScreeningRulesListStatus=<NfScreeningRulesListStatus>&nfScreeningRulesListType=<NfScreeningRulesListType>

Table 6-1 (Cont.) Service API Interfaces

Resource Name	Resource URI	HTTP Method	Data Model for Request	Data Model for Response	Description
screening-rules	{apiRoot}/nrf-configuration/v1/screening-rules/{nfScreeningRulesListType}	PUT	NfScreeningRules	NfScreeningRules	Replace the complete specified NF Screening Rule List Type.
screening-rules	{apiRoot}/nrf-configuration/v1/screening-rules/{nfScreeningRulesListType}	PATCH	PatchDocument	NfScreeningRules	Partially updates the specified NF Screening Rule List Type (except read only attributes).

Responses Supported by Service API Interfaces

Table 6-2 Response Body

Data Type	Presence	Cardinality	Response Codes	Description
ProblemDetails	C	1	500 Internal Server Error	Internal error occurred while processing the service API.
ProblemDetails	C	1	400 Bad Request	JSON body sent by client is not correct according to data model defined.
As per Data Model Defined	C	1	200 OK	Response body contains the store values currently with OCNRF.

Common Data types and Data Models

Common data types

Table 6-3 Common Data Types

DataType	Reference
NFType	3GPP TS 29.510
NFServiceVersion	3GPP TS 29.510
UriScheme	3GPP TS 29.510
Fqdn	3GPP TS 29.510
Ipv6Addr	3GPP TS 29.571
Ipv4Addr	3GPP TS 29.571
Ipv4AddressRange	3GPP TS 29.510
PlmnId	3GPP TS 29.571
Uri	3GPP TS 29.571
IpEndPoint	3GPP TS 29.510
NFType	3GPP TS 29.510

Table 6-3 (Cont.) Common Data Types

DataType	Reference
ProblemDetails	3GPP TS 29.571

Table 6-4 NfConfig

Attribute	DataType	Presence	Description
apiVersions	array (NFServiceVersion)	M	API Version of NF
scheme	UriScheme	M	URI schema supported by NF
fqdn	Fqdn	M	FQDN of NF
port	integer	O	Port of NF default value: 80 if scheme is HTTP, 443 if scheme is HTTPS
apiPrefix	string	O	ApiPrefix
priority	integer	M	Priority of NF
nfInstanceId	string	M	NF Instance Id of NF

Table 6-5 ErrorInfo

Attribute	DataType	Presence	Description
errorCondition	ErrorCondition	ReadOnly	Error Conditions
responseCode	integer	M	This response code is used when corresponding error condition occurs.
errorResponse	string	M	This response description is used when corresponding error condition occurs.
retryAfter	string	C	The attribute indicates the time interval after which the NF shall retry the request. retryAfter header by the OCNRF will be added only for responseCodes - 503, 413, 429, 3xx The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively. Range: 60s-1h Default Value: 5m

Table 6-5 (Cont.) ErrorInfo

Attribute	DataType	Presence	Description
redirectUrl	string	C	The attribute indicates the NF to redirect its request to this uri. location header by the OCNRF will be added only responseCodes - 3xx. redirectUrl should be in URI format. It is mandatory to configure redirectUrl when responseCodes configured.

Table 6-6 ErrorCondition

ErrorCondition	Description
SLF_Missing_Mandatory_Parameters	SLF mandatory parameters are missing
SLF_Not_Reachable	SLF is not reachable from OCNRF
SLF_Subscriber_Not_Provisioned	Subscriber not provisioned in SLF
SLF_OAuthToken_Failure	SLF OauthToken Failure Occurred
NRF_Not_Reachable	NRF not reachable
NRF_Forwarding_Loop_Detection	Loop Detected
RequesterNf_Unauthorized	RequesterNfType is not authorized to receive access token for the targetNfType
Nf_Fqdn_Authentication_Failure	Consumer NF authentication using FQDN failed
Invalid_Key_Details	Configured Key ID details are invalid and cannot be used
Current_Key_Id_Not_Configured	Current Key ID is not configured

Table 6-7 ResponseHttpStatusCodes

Attribute	DataType	Description
pattern	string	Either pattern or codeList is present
codeList	array (integer)	Either pattern or codeList is present

Table 6-8 ScreeningRulesResult

Attribute name	DataType	Presence	Cardinality	Description
nfScreeningRulesList	array (NfScreeningRules)	M	0..N	It contains an array of NF Screening List. An empty array means there is no NF Screening list configured.

General Options

Configuration Example

API:- *{apiRoot}/nrf-configuration/v1/generalOptions*

Method: PUT

Content Type: application/json

Body:

```
{
    "nrfPlmnList": [{
        "mcc": "310",
        "mnc": "14"
    }],
    "ocnrfHost": "ocnrf-ingressgateway.ocnrf.svc.cluster.local",
    "ocnrfPort": 80,
    "enableF3": true,
    "enableF5": true,
    "maximumHopCount": 3,
    "defaultLoad": 5,
    "defaultPriority": 100,
    "defaultPriorityAssignment": false,
    "defaultLoadAssignment": false
}
```

Configuration Attributes

Note:

- If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. Atleast one attribute must be included during PUT request.
- nrfPlmnList, ocnrfHost and ocnrfPort are mandatory values to be configured before using OCNRF.

Table 6-9 GeneralOptions

Attribute Name	DataType	Constraints	Default Value	Description
nrfPlmnList	array (PlmnId)			This value will have at least one PLMN supported by OCNRF and this value shall be set before using OCNRF.

ocnrfHost	string	None	ocnrf-ingressgateway.ocnrf.svc.cluster.local	ocnrfHost needs to be OCNRF's External Routable FQDN (for example, ocnrf.oracle.com) OR External Routable IpAddress (for example, 10.75.212.60) OR for routing with in the same K8 cluster use full OCNRF Ingress Gateway's Service FQDN as below format:
				
				`<helm-releasename>-ingressgateway.<namespace>.svc.<cluster-domainname>`
				
				Example:
				
				`ocnrfingressgateway.nrf-1.svc.cluster.local`
				
				where ocnrf: is the helm release name (deployment name that will be used during "helm install")
				nrf-1: is the namespace in which NRF will be deployed
				cluster.local: is the K8's dnsDomain name (dnsDomain can be found using `kubectl -n kube-system get configmap kubeadmconfig -o yaml | grep -i dnsDomain`)
				This value is used in UriList of NfListRetrieval Service Operation response.

ocnrfPort	integer	None	80	OCNRF Host's Port

Table 6-9 (Cont.) GeneralOptions

Attribute Name	DataType	Constraints	Default Value	Description
enableF3	boolean	true or false	true	OCNRF functions as per 29510 v15.3 specification, if this flag is set to true. If it is set to true, then OCNRF will compliant to 29510 v15.3. If it is set to false, OCNRF will compliant to 29510 v15.2.
enableF5	boolean	true or false	true	OCNRF functions as per 29510 v15.5 specification, if this flag is set to true. If it is set to false, OCNRF functions as per 29510 v15.2 or v15.3 specification (depends on enableF3 flag).
defaultLoad	integer	0 - 100	5	defaultLoad value is set in NF load attribute of NFProfile, if this attribute is set to true. This value is sent in NFDiscover response and NFProfile sent in NFNotify operation, in case NFProfile does not have load attribute.
defaultPriority	integer	0 - 65535	100	This attribute is default value of NF Priority and will be used if NFProfile does not have priority attribute set by NF.
defaultLoadAssignment	boolean	true or false	false	Value of default NF load will be set in NF Load attribute of NFProfile while sending in NFDiscover response and NFProfile sent in NFNotify operation, in case NFProfile does not have Load attribute.
defaultPriorityAssignment	boolean	true or false	false	Value of default NF Priority will be set in NF Priority attribute of NFProfile while sending in NFDiscover response and NFProfile sent in NFNotify operation, in case NFProfile does not have Priority attribute.
maximumHopCount	integer	1-5	3	Maximum number of Nodes (SLF/NRF's) that OCNRF can communicate, to service a request.

NF Screening Options

Configuration Example

API:- *{apiRoot}/nrf-configuration/v1/nfScreeningOptions*

Method: PUT

Content Type: application/json

Body:

```
{
   "featureStatus": "DISABLED",
   "responseCode": 403
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. At-least one attribute shall be included during PUT request.

Table 6-10 nfScreeningOptions

Attribute Name	DataType	Constraints	Default Value	Description
featureStatus	string	ENABLED DISABLED	DISABLED	This attribute indicates if NF Screening Feature is enabled or not globally
responseCode	integer		403	This attribute indicates HTTP status code which will be returned if incoming request does not qualify NF Screening rules

NF Management Options

Configuration Example

API:- *{apiRoot}/nrf-configuration/v1/nfManagementOptions*

Method: PUT

Content Type: application/json

Body:

```
{
    "nfHeartBeatTimers": [{
            "nfType": "ALL_NF_TYPE",
            "minHbTimer": "30s",
            "maxHbTimer": "5m",
            "defaultHbTimer": "30s",
            "nfHeartBeatMissAllowed": 3
        },
        {
            "nfType": "AMF",
            "minHbTimer": "10s",
            "maxHbTimer": "120s",
            "defaultHbTimer": "20s",
            "nfHeartBeatMissAllowed": 1
        }
    ],
    "nfNotifyLoadThreshold": 5,
    "nrfSupportForProfileChangesInResponse": true,
    "defaultSubscriptionValidityTime": "24h",
    "nrfSupportForProfileChangesInNotification": false,
    "nfProfileSuspendDuration": "168h",
    "acceptAdditionalAttributes": false,
    "allowDuplicateSubscriptions": true
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. At-least one attribute shall be included during PUT request.

Table 6-11 nfManagementOptions

Attribute Name	DataType	Constraints	Default Values	Description
nfHeartbeatTimers	array (HeartbeatInfo)	See HeartbeatInfo table for details		This attribute is used to configure the heartbeat related information of the NF. It allows to configure the heartbeat information per NFType. By default, the nfHeartbeatTimer information for ALL_NF_TYPE is present.

Table 6-11 (Cont.) nfManagementOptions

Attribute Name	DataType	Constraints	Default Values	Description
nfNotifyLoadThreshold	integer	0 - 99	5	OCNRF generates the Notification trigger when difference between the 'load' value reported by NF in most recent heartbeat and the last reported ‘load’ is more than configured value of nfNotifyloadThreshold attribute.
nrfSupportForProfileChangesInResponse	boolean	true or false	true	OCNRF sends mandatory and modified attributes in the NFRegister and NFUpdate responses instead of complete profile, if this flag is enabled.
defaultSubscriptionValidityTime	string	10s - 720h	24h	If Validity time attribute is not received in SubscriptionData during NFStatusSubscribe, this default value will be used for calculation of validity time (current time + default duration). If Validity time attribute is received in SubscriptionData during NFStatusSubscribe, this is minimum value will be used for validation and limit purpose. It means if value provided is less than (current time + minimum possible range value), then minimum range value will be considered as validity time for subscription and similarly in case validity time is more than (current time + maximum possible range value), then maximum range value will be considered as validity time for subscription. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.
nrfSupportForProfileChangesInNotification	boolean	true or false	false	OCNRF sends profileChanges attribute instead of NFProfile in Notification, if this flag is enabled.
nfProfileSuspendDuration	string	10s - 744h	168h	Indicates the duration for which the NF is suspended, before it is deleted from OCNRF database. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes and seconds respectively.
acceptAdditionalAttributes	boolean	true or false	false	OCNRF preserves additional attributes that are not defined by 3GPP in NFProfile/NFService based on this attribute value.

Table 6-11 (Cont.) nfManagementOptions

Attribute Name	DataType	Constraints	Default Values	Description
allowDuplicateSubscriptions	boolean	true or false	true	This attribute specifies if OCNRF should allow duplicate Subscriptions to be created or not. Note: In case duplicate subscriptions are not allowed and this flag is marked as false, there will be performance degradation around 50% during NFStatusSubscribe service operation.

HeartbeatInfo

Table 6-12 HeartbeatInfo

Attribute	DataType	Constraints	Description
nfType	string	NFType	All nftypes supported in 29.510 Rel 15.5. In addition to this, *ALL_NF_TYPE* and *CUSTOM_NF_TYPE* shall also be supported. *ALL_NF_TYPE* is the NF Type to be used to specify the default configuration that is to be used when nfType specific configuration is not present. **Notes**: • By Default record will pre-loaded for *ALL_NF_TYPE*. See table "Table 6-13" for details • ALL_NF_TYPE element can't be deleted • *CUSTOM_NF_TYPE* is the NFType to be used to specify the configuration for custom NF types
minHbTimer	string	10s-24h	The minimum HeartbeatTimer allowed for the NF The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.
maxHbTimer	string	10s-24h	The maximum HeartbeatTimer allowed for the NF. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.

Table 6-12 (Cont.) HeartbeatInfo

Attribute	DataType	Constraints	Description
defaultHbTimer	string	minHbTimer and maxHbTimer attributes	This default heartBeatTimer value to be used when the network functions does not provide the heartBeatTimer value in NFProfile. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.
nfHeartbeatMissAllowed	Integer	0-15	The allowed number of missed HeartBeat(s) after which the NFProfile is marked as suspended. If the value is set to 0, NF profiles for which even single heartbeat is missed will be marked as suspended.

Table 6-13 HeartbeatInfo Default Loaded Data

Attribute	Default Loaded Value
nfType	ALL_NF_TYPE
minHbTimer	30s
maxHbTimer	5m
defaultHbTimer	30s
nfHeartbeatMissAllowed	3

NF Discovery Options

Configuration Example

API:- *{apiRoot}/nrf-configuration/v1/nfDiscoveryOptions*

Method: PUT

Content Type: application/json

Body:

```
{
  "profilesCountInDiscoveryResponse": 3,
  "discoveryResultLoadThreshold": 0,
  "discoveryValidityPeriodCfg": [        {
            "nfType": "ALL_NF_TYPE",
            "validityPeriod": "1h"
       },
        {
            "nfType": "AMF",
            "validityPeriod": "3h"
        },
        {
```

```
            "nfType": "CUSTOM_NF_TYPE",
            "validityPeriod": "2h"
        }
    ]
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. At-least one attribute shall be included during PUT request.

Table 6-14 nfDiscoveryOptions

Attribute Name	DataType	Constraints	Default Values	Description
profilesCountInDiscoveryResponse	integer	0 - 20	3	This value restricts NF profile count in NFDiscover response. If value of this attribute is 0, it means this functionality is disabled, in that case all of the NF profiles after the discovery filtering will be returned in NFDiscover response. **Note**: If Limit attribute is present in SearchData URI, then this attribute is not used.
discoveryResultLoadThreshold	integer	0 - 100	0	This configuration is used to select out profiles from discovery response whose load is more than the configured value. NFDiscover response contains NF profiles with load attribute value less than or equal to this configured value. Value 0 indicates this feature is disabled.
discoveryValidityPeriodCfg	array (DiscoveryValidityPeriodCfg)	See DiscoveryValidityPeriodCfg table for details		This attribute mentions the validity period of a discovery request for a specific target-nf-type after which requester NF must perform discovery again to get the latest values. By default, the validityPeriod information for ALL_NF_TYPE is present.

Table 6-15 DiscoveryValidityPeriodCfg

Attribute	DataType	Constraints	Description
nfType	string	NFType	All nftypes supported in 29.510 Rel 16.3.0. In addition to this, *ALL_NF_TYPE* and *CUSTOM_NF_TYPE* shall also be supported. *ALL_NF_TYPE* is the NF Type to be used to specify the default configuration that is to be used when nfType specific configuration is not present. **Notes**: • By Default record will pre-loaded for *ALL_NF_TYPE*. See table "**NfTypeValidityPeriod Loaded Data**" for details • ALL_NF_TYPE element can't be deleted. • *CUSTOM_NF_TYPE* is the NFType to be used to specify the configuration for custom NF types
validityPeriod	string	0s - 720h	This attribute mentions the validity period of a discovery request of a specific target-nf-type after which requester NF must perform discovery again to get the latest values. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.

Table 6-16 NfTypeValidityPeriod Loaded Data

Attribute	Default Loaded Value
nfType	ALL_NF_TYPE
validityPeriod	1h

NF Access Token Options

Configuration Example

API:- *{apiRoot}/nrf-configuration/v1/nfAccessTokenOptions*

Method: PUT

Content Type: application/json

Body:

```
{
   "oauthTokenExpiryTime": "1h",
   "authorizeRequesterNf": "ENABLED",
   "logicalOperatorForScope": "AND",
   "audienceType": "NF_INSTANCE_ID",
   "authFeatureConfig": {
      "featureStatus": "DISABLED",
      "authRulesConfig": [
            {
               "targetNfType":"PCF",
```

```
                "requesterNfType":"AMF",
                "serviceNames":[
                   "npcf-am-policy-control",
                   "npcf-eventexposure"
                ]
             },
             {
                "targetNfType":"UDM",
                "requesterNfType":"AMF",
                "serviceNames":[
                   "*"
                ]
             }
          ],
       "errorResponses": [{
          "errorCondition": "RequesterNf_Unauthorized",
          "responseCode": 400,
          "errorResponse": "The Consumer NfType is not authorized to
receive access token for the requested Nftype.",
          "retryAfter": "5m",
          "redirectUrl": ""
      }]
   },
   "tokenSigningDetails": {
        "currentKeyID": "a14ef8e1bc5c",
        "addkeyIDInAccessToken": true,
        "defaultK8SecretDetails": {
            "k8SecretName": "ocnrf",
            "k8SecretNameSpace": "ocnrf"
        },
        "keyDetailsList": [
            {
                "keyID": "a14ef8e1bc5c",
                "algorithm": "ES256",
                "privateKey": {
                    "k8SecretName": "ocnrf",
                    "k8SecretNameSpace": "ocnrf",
                    "fileName": "ec_private_key_pkcs8.pem"
                },
                "certificate": {
                    "k8SecretName": "ocnrf",
                    "k8SecretNameSpace": "ocnrf",
                    "fileName": "ecdsa_ocnrfapigatewayTestCA.cer"
                }
            }
        ]
    },
   "errorResponses": [
        {
            "errorCondition": "Invalid_Key_Details",
            "responseCode": 500,
            "errorResponse": "Configured Key ID details are invalid and
cannot be used",
            "retryAfter": "5m",
            "redirectUrl": ""
```

```
        },
        {
            "errorCondition": "Current_Key_Id_Not_Configured",
            "responseCode": 500,
            "errorResponse": "Current Key ID is not configured",
            "retryAfter": "5m",
            "redirectUrl": ""
        }
    ]
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. At-least one attribute shall be included during PUT request.

Table 6-17 nfAccessTokenOptions

Attribute Name	DataType	Constraints	Default Values	Description
oauthTokenExpiryTime	string	1s - 168h	1h	Oauth token expiry time. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes and seconds respectively. **Note**: In case OCNRF signed certificate expiry duration is less than this attribute, then certificate expiry duration is used in Access Token Expiry Time.
authorizeRequesterNf	string	ENABLED, DISABLED	ENABLED	This attribute validates the requester NF is registered with OCNRF or not. OCNRF issue the access token only to the registered requester NFs. If NF is registered, then check if NFtype in Access Token Request is same as in NF profile registered with NRF and requesterPlmn received in the Access Token Request request is same as Registered Profile. If the value is Disabled, OCNRF will issue token to non-registered NFs as well.

Table 6-17 (Cont.) nfAccessTokenOptions

Attribute Name	DataType	Constraints	Default Values	Description
audienceType	string	NF_INSTANCE_ID, NF_TYPE	NF_INSTANCE_ID	This value decides the AudienceType in AccessTokenClaim. OCNRF considers this value only if targetnfInstanceId is not received in AccessTokenRequest. NF_INSTANCE_ID - NF Instance Id(s) in audience IE of AccessTokenClaim NF_TYPE - NF Type in audience IE of AccessTokenClaim
logicalOperatorForScope	string	AND, OR	AND	This value will decide whether values in scope will have relationship AND or OR. If value is AND, while looking for producer network function profiles, token will be issued for profiles matching all the services-names present in scope. If value is OR, token will be issued for profiles matching any of the services-names present in scope.
authFeatureConfig	AuthFeatureConfig	See AuthFeatureConfig table for details		The attribute contains the parameters required to enable and configure NfAccessToken Authorization Feature
tokenSigningDetails	TokenSigningDetails	See TokenSigningDetails for details	null (None of token signing details are configured)	This attribute allows user to configure all of the details required to sign the token generated by OCNRF
errorResponses	array(ErrorInfo)	See table Preloaded Values for details		This attribute allows user to update details for different error conditions

Table 6-18 PreLoaded records for errorResponses

errorCondition	responseCode	errorResponse	retryAfter	redirectUrl
Invalid_Key_Details	500	Configured Key ID details are invalid and cannot be used	5m	
Current_Key_Id_Not_Configured	500	Current Key ID is not configured	5m	

AuthFeatureConfig

Table 6-19 AuthFeatureConfig

Attribute	Data Type	Constraints	Default Value	Description
featureStatus	string	ENABLED, DISABLED	DISABLED	Enables/Disables the NfAccessToken Authorization Feature.
authRulesConfig	array (AuthConfig)			The attribute defines a mapping across Requester NF Type, Target NF Type and the allowed Services. This attribute should be configured if the authFeatureStatus is set to 'ENABLED' Refer Note.
errorResponses	array (ErrorInfo)		See PreLoaded records for AuthFeatureConfig errorResponses below	This attribute defines the error responses which may be sent during NRF AccessToken Authorization failure scenarios. This attribute will allow to update the error response code and error response description. This attribute should be configured if the authFeatureStatus is set to 'ENABLED'. By default, the RequesterNF_Unauthorized condition shall be preloaded. Refer Note.

Table 6-20 PreLoaded records for AuthFeatureConfig errorResponses

errorCondition	responseCode	errorResponse	retryAfter	redirectUrl
RequesterNf_Unauthorized	400	The RequesterNfType is not authorized to receive access token for the targetNfType.	5m	

Note:

The attributes featureStatus, authRulesConfig and errorResponses can be configured in any order and independently. However, when the feature is enabled, it is expected that the authRulesConfig is already configured previously or present in the current request.

Table 6-21 AuthConfig

Attribute	Datatype	Constraints	Description
targetNfType	string	NFType	The attribute defines the NF Type of the target NF.
requesterNfType	string	NFType	The attribute defines the NF Type of the requester NF that is authorized to access the target NF Type and its services.
serviceNames	array(string)	None	This attribute defines the NF services that is authorized to be accessed by the requester NF type. The value "*" shall indicate that all the services are authorized to be accessed the requester NF Type. If "*" is to be used, The services shall contain only a single entry in the list with this value.

Note:

All values of this structure are mandatory to be configured all together.

Table 6-22 TokenSigningDetails

Attribute	Datatype	Constraints	Description
currentKeyID	string	Mandatory attribute for Access Token Service to work. Once currentKeyID is configured, this value cannot be null. Newly added KeyID details can be used after values are validated by OCNRF. Newly added KeyID cannot be configured as currentKeyID in same request.	Key ID value corresponding to which token signing details used to sign the token.
addkeyIDInAccessToken	boolean	true, false. (default: false) Value for this attribute cannot be set to true if currentKeyID is not set.	Value of this attribute decides that KeyID value can be added in AccessToken Response or not. If value is true, then currentKeyID value will be added in AccessToken Response. If value is false, then value will not be added in AccessToken Response.
defaultK8SecretDetails	DefaultK8Secret Details	See DefaultK8SecretDetails table for details. This value is mandatory in case secret namespace and secret name of any individual key details are not configured.	This attribute decides the default K8 secret details and these details value are used in case individual key details for secret name and secret namespace are not configured.
keyDetailsList	array(OauthKeyDetails)	See OauthKeyDetails table for details. In case any of key details need to be modified, then complete keyDetailList is used for updates. Change of specific keyId detail is not supported. Maximum 25 key details can be configured.	This attribute provides details of oauth key details which is used by OCNRF to sign the token. Added Keys status can be get using OCNRF configuration status REST APIs. Any key id details cannot be removed if it is getting used as currentKeyID.

Table 6-23 DefaultK8SecretDetails

Attribute	Datatype	Constraints	Description
k8SecretName	string	Mandatory if Default K8 secret Details are configured. Both k8SecretName and k8SecretNameSpace are configured together.	Default Kubernetes secret name
k8SecretNameSpace	string	Mandatory if Default K8 secret Details are configured. Both k8SecretName and k8SecretNameSpace are configured together.	Default Kubernetes secret namespace

Table 6-24 OauthKeyDetails

Attribute	Datatype	Constraints	Description
keyID	string	Mandatory attribute, keyID length shall not exceed 36 characters	Unique value in list of keys. Key Details are known by this value.
algorithm	string	Mandatory attribute, algorithm can be only ES256 and RS256	algorithm value will be used to sign the oauth token.
privateKey	OauthSecretFiles	Mandatory attribute, see OauthSecretFiles for details	OCNRF Private key details. Both k8SecretName and k8SecretNameSpace are configured together.
certificate	OauthSecretFiles	Mandatory attribute, see OauthSecretFiles for details	OCNRF Public certificate details. Both k8SecretName and k8SecretNameSpace are configured together.

Table 6-25 OauthSecretFiles

Attribute	Datatype	Constraints	Description
k8SecretName	string	Optional, but if this attribute is present then k8SecretNameSpace shall be present	Kubernetes secret name where key and certificate details are stored
k8SecretNameSpace	string	Optional, but if this attribute is present then k8SecretName shall be present	Kubernetes namespace for secret name

Table 6-25 (Cont.) OauthSecretFiles

Attribute	Datatype	Constraints	Description
fileName	string	Mandatory attribute	Filename of Key/ certificate

NRF-NRF Forwarding Options

Configuration Example

API:- *{apiRoot}/nrf-configuration/v1/forwardingOptions*

Method: PUT

Content Type: application/json

Body:

```
{
    "profileRetrievalStatus": "DISABLED",
    "subscriptionStatus": "DISABLED",
    "discoveryStatus": "DISABLED",
    "accessTokenStatus": "DISABLED",
    "nrfHostConfig": [{
        "nfInstanceId": "c56a4180-65aa-42ec-a945-5fd21dec0538",
        "apiVersions": [{
            "apiVersionInUri": "v1",
            "apiFullVersion": "15.5.0"
        }],
        "scheme": "http",
        "fqdn": "ocnrf-1-ingressgateway.ocnrf.svc.cluster.local",
        "priority": 100,
        "port": 80
    }],
    "nrfRerouteOnResponseHttpStatusCodes": {
        "pattern": "^[3,5][0-9]{2}$"
    },
    "errorResponses": [{
            "errorCondition": "NRF_Not_Reachable",
            "responseCode": 504,
            "errorResponse": "NRF not reachable",
            "retryAfter": "5m",
            "redirectUrl": ""
        },{
            "errorCondition": "NRF_Forwarding_Loop_Detection",
            "responseCode": 508,
            "errorResponse": "Loop Detected",
            "retryAfter": "5m",
            "redirectUrl": ""
    }]
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. At-least one attribute shall be included during PUT request.

Table 6-26 forwardingOptions

Attribute Name	DataType	Constraints	Default Values	Description
nrfHostConfig	array (NFConfig)			This is used to configure Primary and Secondary NRF Details which is used for forwarding various requests. It allows to configure details of NRF like apiVersion, scheme, FQDN, port, etc. The only supported value for apiVersionInUri is v1. Hence the apiVersions attribute must have at least one data record with apiVersionInUri attribute values set as v1. This configuration allows you to configure more than two NRF Details. NRF with highest priority is considered as Primary NRF for forwarding messages. NRF with second highest priority is considered as Secondary NRF for forwarding. To reset this attribute, please send empty array, for example:- "nrfHostConfig": [ ] If this attribute is already set then there is no need to provide the value again.
nrfRerouteOnResponseHttpStatusCodes	ResponseHttpStatusCodes	pattern or specific code list	"pattern": "^[3,5][0-9]{2}$"	This configuration is used to determine if the service operation message needs to forwarded to Secondary NRF. After getting response from primary NRF, if response status code from primary NRF matches with the configured response status code list, then NRF reroutes the request to the secondary NRF. Refer nrfHostConfig for details for Primary and Secondary NRF details.

Table 6-26 (Cont.) forwardingOptions

Attribute Name	DataType	Constraints	Default Values	Description
profileRetrievalStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the forwarding of NFProfileRetrieval service operation messages. If the flag is set to true and OCNRF is not able to complete the request due to unavailability of any matching profile, then OCNRF forwards the NfProfileRetrieval request to the configured NRF host(s) and relays the response received from forwarding NRF to the Consumer NF. If flag is false, OCNRF will not forward the NfProfileRetrieval request in any case. It will return a response to consumer NF without forwarding it.
subscriptionStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the forwarding of NFStatusSubscribe, NFStatusUnsubscribe service operation messages. If the flag is set to true and OCNRF is not able to complete the request due to unavailability of any matching profile, then OCNRF forwards the NfStatusSubscribe/ NfStatusUnSubscribe request to the configured NRF host(s) and relays the response received from forwarding NRF to the Consumer NF. If flag is false, OCNRF will not forward the NFStatusSubscribe/ NFStatusUnSubscribe request in any case. It will return a response to consumer NF without forwarding it. **Note**: NFStatusSubscribe forwarding is supported only if Subscription Condition is NfInstanceIdCond in the NFStatusSubscribe request.

Table 6-26 (Cont.) forwardingOptions

Attribute Name	DataType	Constraints	Default Values	Description
discoveryStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the forwarding of NFDiscover service operation messages. If the flag is set to true and OCNRF is not able to complete the request due to unavailability of any matching profile, then OCNRF forwards the NFDiscover request to the configured NRF host(s) and relays the response received from forwarding NRF to the Consumer NF. If flag is false, OCNRF will not forward the NFDiscover request in any case. It will return a response to consumer NF without forwarding it.
accessTokenStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the forwarding of AccessToken service operation messages. If the flag is set to true and OCNRF is not able to complete the request due to unavailability of any matching Producer NF, then OCNRF forwards the AccessToken request to the configured NRF host(s) and relays the response received from forwarding NRF to the Consumer NF. If flag is false, OCNRF will not forward the AccessToken request in any case. It will return a response to consumer NF without forwarding it.
errorResponses	array (ErrorInfo)		See PreLoaded values below	This attribute defines the error responses which may be sent during NRF Forwarding scenarios. This attribute will allow to update the error response code and error response description for preloaded error conditions.

Table 6-27 PreLoaded records for errorResponses

errorCondition	responseCode	errorResponse	retryAfter	redirectUrl
NRF_Not_Reachable	504	NRF not reachable	5m	
NRF_Forwarding_Loop_Detection	508	Loop Detected	5m	

SLF Options

Configuration Example

API:- *{apiRoot}/nrf-configuration/v1/slfOptions*

Method: PUT

Content Type: application/json

Body:

```
{
    "supportedNfTypeList": [],
        "preferredSubscriberIdType": "SUPI",
        "slfHostConfig": [{
            "nfInstanceId": "c56a4180-65aa-42ec-a945-5fd21dec0538",
            "apiVersions": [{
                "apiVersionInUri": "v1",
                "apiFullVersion": "15.5.0"
            }],
            "scheme": "http",
            "fqdn": "ocudrSlf-1-ingressgateway.ocnrf.svc.cluster.local",
            "priority": 100,
            "port": 80
        }],
        "rerouteOnResponseHttpStatusCodes": {
            "codeList": [134]
        },
        "featureStatus": "DISABLED",
        "useOAuthToken": false,
        "errorResponses": [{
            "errorCondition": "SLF_Missing_Mandatory_Parameters",
            "responseCode": 400,
            "errorResponse": "Mandatory parameter missing for SLF
Lookup",
            "retryAfter": "5m",
            "redirectUrl": ""
        }, {
            "errorCondition": "SLF_Subscriber_Not_Provisioned",
            "responseCode": 200,
            "errorResponse": "Subscriber not provisioned in SLF",
            "retryAfter": "5m",
            "redirectUrl": ""
        }, {
            "errorCondition": "SLF_Not_Reachable",
            "responseCode": 504,
            "errorResponse": "SLF not reachable",
            "retryAfter": "5m",
            "redirectUrl": ""
        },
        {
            "errorCondition": "SLF_OAuthToken_Failure",
            "responseCode": 500,
```

```
            "errorResponse": "SLF OauthToken Failure Occurred",
            "retryAfter": "5m",
            "redirectUrl": ""
        }]
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. At-least one attribute shall be included during PUT request.

Table 6-28 slfOptions

Attribute Name	DataType	Constraints	Default Values	Description
featureStatus	string	ENABLED, DISABLED	DISABLED	Enables/disables the SLF Feature. • In case GroupId is already present in Search Query, even value of this attribute is ENABLED. OCNRF will use the Group Id received and will not communicate to SLF/UDR. • If Subscriber Id is present in Search Query it will be ignored and will not be used to perform discovery search in OCNRF.

Table 6-28 (Cont.) slfOptions

Attribute Name	DataType	Constraints	Default Values	Description
slfHostConfig	array (NFConfig)			This is used to configure Primary and Secondary SLF Details which is used for forwarding various requests. It allows to configure details of SLF like apiVersion, scheme, FQDN, port, etc. The only supported value for apiVersionInUri is v1. Hence the apiVersions attribute must have at least one data record with apiVersionInUri attribute values set as v1. This configuration allows you to configure more than 2 SLF Details. SLF with highest priority is considered as Primary SLF for forwarding messages. SLF with second highest priority is considered as Secondary SLF for forwarding. If supportedNfTypeList is set, then operator must set this attribute. This is because this value will be used to contact the network function hosting the SLF. To reset this attribute, please send empty array, for example:- "slfHostConfig": [ ] If this attribute is already set then there is no need to provide the value again.
supportedNfTypeList	array (NfType)			NF Type list for which SLF need to be supported. SLF look up will happen only for NF Types mentioned in this configuration. OCNRF supports the below NF Type(s) for performing SLF lookup: • UDM • UDR • AUSF To reset this attribute, send empty array, for example: "supportedNfTypeList": [ ] If this value is set, then slfHostConfig shall also be set.

Table 6-28 (Cont.) slfOptions

Attribute Name	DataType	Constraints	Default Values	Description
preferredSubscriberIdType	string	SUPI or GPSI	SUPI	This attribute will only be used, in case different type of subscriber identifiers (SUPI, GPSI) are present in NFDiscover service operation message, which subscriber identifier shall be used for the query to SLF
rerouteOnResponseHttpStatusCodes	ResponseHttpStatusCodes	pattern or codeList. See table ResponseHttpStatusCodes for details	"pattern": "^[3,5][0-9]{2}$"	This attribute will be used after getting response from primary SLF (SLF Config with highest priority), if response code from primary SLF is present/matches this configuration, then OCNRF will reroute the SLF query to secondary SLF (SLF Config with second highest priority).
useOAuthToken	boolean	true or false	false	This attribute is used while doing SLF query to UDR/SLF to query the Access Token Service of OCNRF to get Oauth access token details for UDR/SLF network function to which query will be done. If value of this attribute is true, SLF function of OCNRF access the Access Token Service. If value of this attribute is false, SLF function will not access the Access Token Service and will do SLF query without Oauth Access Token.
errorResponses	array (ErrorInfo)		See PreLoaded values below	This attribute defines the error responses which may be sent during SLF processing. This attribute will allow to update the error response code and error response description for preloaded error conditions.

Table 6-29 PreLoaded records for errorResponses

errorCondition	responseCode	errorResponse	retryAfter	redirectUrl
SLF_Missing_Mandatory_Parameters	400	Mandatory parameter missing for SLF Lookup	5m	
SLF_Not_Reachable	504	SLF not reachable	5m	
SLF_Subscriber_Not_Provisioned	200	Subscriber not provisioned in SLF	5m	

Table 6-29 (Cont.) PreLoaded records for errorResponses

errorCondition	responseCode	errorResponse	retryAfter	redirectUrl
SLF_OAuthToken_Failure	500	SLF OauthToken Failure Occurred	5m	

Geo Redundancy Options

Configuration Example
API: *{apiRoot}/nrf-configuration/v1/geoRedundancyOptions*

Method: PUT

Content Type: application/json

Body:

```
{
  "geoRedundancyFeatureStatus": "DISABLED",
  "replicationLatency": "5s",
  "monitorNrfServiceStatusInterval": "5s",
  "monitorDBReplicationStatusInterval": "5s"
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. Atleast one attribute shall be included during PUT request.

Table 6-30 geoRedundancyOptions

Attribute Name	Data Type	Constraints	Default Values	Description
featureStatus	string	ENABLED, DISABLED	DISABLED	Enables/Disables the geoRedundancy feature in OCNRF.
replicationLatency	string	1s - 10m	5s	This attribute defines the time taken for the data in the database to get replicated between GeoRedundant OCNRFs. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.

Table 6-30 (Cont.) geoRedundancyOptions

Attribute Name	Data Type	Constraints	Default Values	Description
monitorNrfServiceStatusInterval	string	1s - 10s	5s	This attribute defines the time interval for monitoring the aggregated Nf_Management service status (combined status of nfRegistration, nfSubscription and nrfAuditor service). The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.
monitorDBReplicationStatusInterval	string	1s - 10s	5s	This attribute defines the time interval for monitoring the DB replication status. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.

NF Authentication Options

Configuration Example
API: *{apiRoot}/nrf-configuration/v1/nfAuthenticationOptions*

Method: PUT

Content Type: application/json

Body:

```
{
    "nfRegistrationStatus": "DISABLED",
    "nfSubscriptionStatus": "DISABLED",
    "nfDiscoveryStatus": "DISABLED",
    "accessTokenStatus": "DISABLED",
    "nfProfileRetrievalStatus": "DISABLED",
    "nfListRetrievalStatus": "DISABLED",
    "checkIfNfIsRegistered": "DISABLED",
    "errorResponses": [{
    "errorCondition": "Nf_Fqdn_Authentication_Failure",
       "responseCode": 403,
       "errorResponse": "Failed to authenticate NF using FQDN",
       "retryAfter": "5m",
       "redirectUrl": ""
   }]
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. At least one attribute shall be included during PUT request.

Table 6-31 nfAuthenticationOptions

Attribute Name	Data Type	Constraints	Default Values	Description
nfRegistrationStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the authentication of consumer NF for NFRegister, NFUpdate and NFDeregister service operations. If value of this attribute is ENABLED, then identity of consumer NF is validated. If value of this attribute is DISABLED, then validation is not performed for consumer NF.
nfSubscriptionStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the authentication of consumer NF for NFStatusSubscribe and NFStatusUnsubscribe service operations. If value of this attribute is ENABLED, then identity of consumer NF is validated and NRF allows the subscription only if the NF is registered with OCNRF. If value of this attribute is DISABLED, then validation is not performed for consumer NF.
nfDiscoveryStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the authentication of consumer NF for NFDiscover service operation. If value of this attribute is ENABLED, then identity of consumer NF is validated. If value of this attribute is DISABLED, then validation is not performed for consumer NF. In case NF identity is not present in discovery request messages then validation is performed as per checkIfNfIsRegistered attribute.

Table 6-31 (Cont.) nfAuthenticationOptions

Attribute Name	Data Type	Constraints	Default Values	Description
accessTokenStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the authentication of consumer NF for AccessToken service operation. If value of this attribute is ENABLED, then identity of consumer NF is validated. If value of this attribute is DISABLED, then validation is not performed for consumer NF.
nfProfileRetrievalStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the authentication of consumer NF for NF Profile Retrieval service operation. If value of this attribute is ENABLED, then identity of consumer NF is validated. If value of this attribute is DISABLED, then validation is not performed for consumer NF.
nfListRetrievalStatus	string	ENABLED, DISABLED	DISABLED	This attribute controls the authentication of consumer NF for NF List Retrieval service operation. If value of this attribute is ENABLED, then identity of consumer NF is validated. If value of this attribute is DISABLED, then validation is not performed for consumer NF.
errorResponses	array (ErrorInfo)		See PreLoaded values below	This attribute defines the error responses which may be sent for NF Authentication scenarios. This attribute will allow to update the response code, error response description, retryAfter and redirectUrl for preloaded error conditions.
checkIfNfIsRegistered	string	ENABLED, DISABLED	DISABLED	This attribute controls the mechanism if there is need to check if NF is registered or not with OCNRF. If value of this attribute is ENABLED, then validation is performed. If value of this attribute is DISABLED, then validation is not performed for consumer NF. • discovery request does not contain requester-nf-instance-fqdn and value of nfDiscoveryAuthenticationStatus is ENABLED.

Table 6-32 PreLoaded records for errorResponses

errorCondition	responseCode	errorResponse	retryAfter	redirectUrl
Nf_Fqdn_Authentication_Failure	403	Failed to authenticate NF using FQDN	5m	

Log Level Options

Configuration Example

API:*{apiRoot}/nrf-configuration/v1/logLevelOptions*

Method: PUT

Content Type: application/json

Body:

```
{
  "nfSubscriptionLogLevel": "WARN",
  "nfRegistrationLogLevel": "WARN",
  "nfDiscoveryLogLevel": "WARN",
  "nfAccessTokenLogLevel": "WARN",
  "nrfAuditorLogLevel": "WARN",
  "nrfConfigurationLogLevel": "WARN"
}
```

Configuration Attributes

> **Note:**
>
> If any attribute is not present in JSON request body while updating, existing value in database will be preserved and used. Atleast one attribute shall be included during PUT request.

Table 6-33 logLevelOptions

Attribute Name	Data Type	Constraints	Default Values	Description
nfSubscriptionLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Application Log Level for the NFSubscription Microservice
nfRegistrationLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Application Log Level for the NFRegistration Microservice
nfDiscoveryLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Application Log Level for the NFDiscovery Microservice

Table 6-33 (Cont.) logLevelOptions

Attribute Name	Data Type	Constraints	Default Values	Description
nfAccessTokenLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Application Log Level for the NFAccessToken Microservice
nrfAuditorLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Application Log Level for the NRFAuditor Microservice
nrfConfigurationLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Application Log Level for the NRFConfiguration Microservice

NF Screening Configuration

Configuration Example

Screening Rules List Update

NF screening rules update particular rule configuration (except read only attributes)

URI: *http://host:port/nrf-configuration/v1/screening-rules/CALLBACK_URI*

Method: PUT

Content Type: application/json

Body:

Request Body

```
{
    "nfScreeningType": "BLACKLIST",
    "nfScreeningRulesListStatus": "ENABLED",
    "globalScreeningRulesData": {
        "failureAction": "SEND_ERROR",
        "nfCallBackUriList": [
            {
                "ipv4AddressRange":{
                    "start": "155.90.171.123",
                    "end": "233.123.19.165"
                },
                "ports":[10,20]
            },
            {
                "ipv6AddressRange":{
                    "start": "1001:cdba:0000:0000:0000:0000:3257:9652",
                    "end": "3001:cdba:0000:0000:0000:0000:3257:9652"
                }
            }
        ]
    },
```

```
    "amfScreeningRulesData": {
        "failureAction": "CONTINUE",
        "nfCallBackUriList": [
            {
                "fqdn": "ocnrf-d5g.oracle.com"
            },
            {
                "ipv4AddressRange":{
                    "start": "155.90.171.123",
                    "end": "233.123.19.165"
                },
                "ports":[10,20]
            }
        ]
    }
}
```

Screening Rules List Get

NF screening rules get all rules
URI: *http://host:port/nrf-configuration/v1/screening-rules/*

Method: GET

Response Body:

```
{
    "nfScreeningRulesList": [
        {
            "nfScreeningRulesListType": "NF_FQDN",
            "nfScreeningType": "BLACKLIST",
            "nfScreeningRulesListStatus": "DISABLED"
        },
        {
            "nfScreeningRulesListType": "NF_IP_ENDPOINT",
            "nfScreeningType": "BLACKLIST",
            "nfScreeningRulesListStatus": "ENABLED",
            "amfScreeningRulesData": {
                "failureAction": "SEND_ERROR",
                "nfIpEndPointList": [
                    {
                        "ipv4Address": "198.21.87.192",
                        "ports": [
                            10,
                            20
                        ]
                    }
                ]
            }
        },
        {
            "nfScreeningRulesListType": "CALLBACK_URI",
            "nfScreeningType": "BLACKLIST",
            "nfScreeningRulesListStatus": "ENABLED",
            "globalScreeningRulesData": {
```

```
                "failureAction": "SEND_ERROR",
                "nfCallBackUriList": [
                    {
                        "fqdn": "ocnrf-d5g.oracle.com",
                        "ports": [
                            10,
                            20
                        ]
                    }
                ]
            }
        },
        {
            "nfScreeningRulesListType": "PLMN_ID",
            "nfScreeningType": "BLACKLIST",
            "nfScreeningRulesListStatus": "DISABLED"
        },
        {
            "nfScreeningRulesListType": "NF_TYPE_REGISTER",
            "nfScreeningType": "WHITELIST",
            "nfScreeningRulesListStatus": "ENABLED",
            "globalScreeningRulesData": {
                "failureAction": "SEND_ERROR",
                "nfTypeList": [
                    "AMF",
                    "SMF",
                    "PCF"
                ]
            }
        }
    ]
}
```

Screening Rules List Partial Update

NF screening rules get particular configured rule
URI: *http://host:port/nrf-configuration/v1/screening-rules/CALLBACK_URI*

Method: GET

Response Body:

```
{
    "nfScreeningRulesListType": "CALLBACK_URI",
    "nfScreeningType": "BLACKLIST",
    "nfScreeningRulesListStatus": "ENABLED",
    "globalScreeningRulesData": {
        "failureAction": "SEND_ERROR",
        "nfCallBackUriList": [
            {
                "ipv4AddressRange": {
                    "start": "155.90.171.123",
                    "end": "233.123.19.165"
                },
                "ports": [
```

```
                    10,
                    20
                ]
            },
            {
                "ipv6AddressRange": {
                    "start": "1001:cdba:0000:0000:0000:0000:3257:9652",
                    "end": "3001:cdba:0000:0000:0000:0000:3257:9652"
                }
            }
        ]
    },
    "amfScreeningRulesData": {
        "failureAction": "SEND_ERROR",
        "nfCallBackUriList": [
            {
                "fqdn": "ocnrf-d5g.oracle.com"
            },
            {
                "ipv4AddressRange": {
                    "start": "155.90.171.123",
                    "end": "233.123.19.165"
                },
                "ports": [
                    10,
                    20
                ]
            }
        ]
    }
}
```

NF screening rules partial rule update
URI: *http://host:port/nrf-configuration/v1/screening-rules/CALLBACK_URI*

Method: PATCH

Content-Type: application/json-patch+json

Request Body:

```
[
    {"op":"remove","path":"/globalScreeningRulesData/
nfCallBackUriList/2/ports/0"},
    {"op":"replace","path":"/globalScreeningRulesData/
failureAction","value": "CONTINUE"},
    {"op":"add","path":"/nrfScreeningRulesData","value":
{"failureAction": "SEND_ERROR","nfCallBackUriList":
      [{"ipv4AddressRange":{"start" : "189.163.192.10","end":
"190.178.127.10"}}]}}
 ]
```

Configuration Attributes

Table 6-34 nfScreeningRules

Attribute name	Data type	Presence	Description
nfScreeningRulesListType	NfScreeningRulesListType	C	ReadOnly. It is returned while retrieving the rule.
nfScreeningType	NfScreeningType	M	Screening type of complete screening list. Blacklist or whitelist. All the rules can be either blacklist or whitelist.
nfScreeningRulesListStatus	NfScreeningRulesListStatus	M	Enables or disables complete screening list.
globalScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if global screening rules need to be configured.
customNfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for custom NF need to be configured.
nrfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for NRF need to be configured.
udmScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for UDM need to be configured.
amfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for AMF need to be configured.
smfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for custom SMF need to be configured.
ausfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for AUSF need to be configured.
nefScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for NEF need to be configured.
pcfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for PCF need to be configured.
nssfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for NSSF need to be configured.
udrScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for UDR need to be configured.
lmfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for IMF need to be configured.
gmlcScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for GMLC need to be configured.

Table 6-34 (Cont.) nfScreeningRules

Attribute name	Data type	Presence	Description
fiveG_EirScreeningRules	NfScreeningRulesData	O	This attribute will be present if screening rules for EIR need to be configured.
seppScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for SEPP need to be configured.
upfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for UPF need to be configured.
n3iwfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for IWF need to be configured.
afScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for AF need to be configured.
udsfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for UDSF need to be configured.
bsfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for BSF need to be configured.
chfScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for CHF need to be configured.
nwdafScreeningRulesData	NfScreeningRulesData	O	This attribute will be present if screening rules for NWDAF need to be configured.

Table 6-35 NfScreeningRulesData

Attribute name	Data type	Presence	Description
failureAction	FailureAction	M	Indicates what action needs to be taken during failure
nfFqdn	NfFqdn	C	This attribute may be present if screeningListType is NF_FQDN
nfCallBackUriList	array(NfCallBackUri)	C	This attribute may be present if screeningListType is CALLBACK_URI
nfIpEndPointList	array(NfIpEndPoint)	C	This attribute shall be present if screeningListType is NF_IP_ENDPOINT
plmnList	array(PlmnId)	C	This attribute shall be present if screeningListType is PLMN_ID
nfTypeList	array(string)	C	This attribute shall be present if screeningListType is NF_TYPE_REGISTER

Table 6-36 NfScreeningRulesListType

Enumeration value	Description
"NF_FQDN"	Screening List type for NF FQDN
"NF_IP_ENDPOINT"	Screening list type for IP Endpoint
"CALLBACK_URI"	Screening list type for callback URIs in NF Service and nfStatusNotificationUri in SubscriptionData
"PLMN_ID"	Screening list type for PLMN ID
"NF_TYPE_REGISTER"	Screening list type for allowed NF Types to register

Table 6-37 NfScreeningType

Enumeration value	Description
"BLACKLIST"	When a screening list is configured to operate as a blacklist, the request is allowed to access the service only if the corresponding attribute value is not present in the blacklist.
"WHITELIST"	When a screening list is configured to operate as a whitelist, the request is allowed to access the service only if the corresponding attribute value is present in the whitelist.

Table 6-38 NfScreeningRulesListStatus

Enumeration value	Description
"ENABLED"	Screening List feature is enabled to apply the rules
"DISABLED"	Screening List feature is disabled

Table 6-39 FailureAction

Enumeration value	Description
"CONTINUE"	Continue Processing
"SEND_ERROR"	Send response with configured HTTP status code

Table 6-40 NfFqdn

Attribute Name	Data Type	Presence	Description
fqdn	array(FQDN)	C	Exact FQDN to be matched. This is conditional, at least one attribute is present.
pattern	array(string)	C	Regular Expression for FQDN. This is conditional, at least one attribute is present.

Table 6-41 NfIpEndPoint

Attribute Name	Data Type	Presence	Description
ipv4Address	Ipv4Addr	C	IPv4 address to be matched. See NOTE below
ipv4AddressRange	Ipv4AddressRange	C	Range of IPv4 addresses. See NOTE below
ipv6Address	Ipv6Addr	C	IPv6 address to be matched. See NOTE below
ipv6AddressRange	Ipv6AddressRange	C	Range of IPv6 addresses. See NOTE below
port	array(integer)	O	If this attribute is not configured then it will not be considered for validation
portRange	array(PortRange)	O	If this attribute is not configured then it will not be considered for validation

Note:

This is conditional, only one attribute must be present (Either of the ipv4Address, ipv4AddressRange, ipv6Address, ipv6AddressRange).

Table 6-42 NfCallBackUri

Attribute Name	Data Type	Presence	Description
fqdn	FQDN	C	Exact Fqdn to be matched. See NOTE below
pattern	string	C	Regular Expression for FQDN, Ipv4Address, Ipv6Address. See NOTE below
ipv4Address	Ipv4Addr	C	IPv4 address to be matched. See NOTE below
ipv4AddressRange	Ipv4AddressRange	C	Range of IPv4 addresses. See NOTE below
ipv6Address	Ipv6Addr	C	IPv6 address to be matched. See NOTE below
ipv6AddressRange	Ipv6AddressRange	C	Range of IPv6 addresses. See NOTE below
port	array(integer)	O	If this attribute is not configured then it will not be considered for validation
portRange	array(PortRange)	O	If this attribute is not configured then it will not be considered for validation

> **Note:**
>
> This is conditional, only one attribute must be present (either of the fqdn, pattern, ipv4Address, ipv4AddressRange, ipv6Address, ipv6AddressRange).

Table 6-43 PortRange

Attribute Name	Data Type	Presence	Description
start	integer	M	First value identifying the start of port range
end	integer	M	Last value identifying the end of port range

Table 6-44 PortRange

Attribute Name	Data Type	Presence	Description
start	Ipv6Addr	M	First value identifying the start of an IPv6 Address range
end	Ipv6Addr	M	Last value identifying the end of an IPv6 Address range

7
OCNRF state data retrieval REST APIs

REST API Details

"`apiRoot`" is concatenation of the following parts

- **scheme**: http, https
- the fixed string "://"
- authority (host and optional port)
  host and port will be CNCC host and port details

Table 7-1 API Details

API	HTTP method supported	Description	HTTP response codes
{apiRoot}/nrf-state-data/v1/nf-details	GET	This API fetches NF Profile related data. Both query request attributes and query result attributes can be used together to get specific results	200 OK with **NFProfileDetails**, if NF Details found 200 OK with Empty List <**NFProfileDetails**>, if NF Details not found 400 BAD Request, if request is not proper 404 NOT FOUND - If no NF Details found for input attributes and query request attribute is used to get details based on specific attribute 500 INTERNAL ERROR - If any internal error occurred while accessing OCNRF state data
{apiRoot}/nrf-state-data/v1/subscription-details	GET	This API fetches Subscription related data. Both query request attributes and query result attributes can be used together to get specific results	200 OK - **SubscriptionDetails** , if subscription details found 200 OK - Empty List <**SubscriptionDetails** >, if subscription details not found 400 BAD Request, if request is not proper 404 NOT FOUND - If no NF Details found for input attributes and query request attribute is used to get details based on specific attribute 500 INTERNAL ERROR - If any internal error occurred while accessing OCNRF state data

Details of nf-details URI

Query request parameters supported by nf-details URI

Note:

- These attributes can be used to get results based on specific input attributes. These attribute are optional to API. In case no query request and result attribute is mentioned, then only NFinstance Ids will be provided for all of the NF Profiles.
- In case complete profile is needed, query request attributes shall contain specific attributes **nf-instance-id** or **nf-fqdn**. Otherwise only NFinstance Ids will be provided for all of the NF Profiles by default and additionally query result attributes can be provided to get additional details.
- At-most one query request attribute is supported, data is returned based on the request attribute.

Table 7-2 Query request attributes supported by nf-details URI

Name	Data Type	Details	Query Example with API
nf-instance-id	string	NF Instance Id of Network Function	{apiRoot}/nrf-state-data/v1/nf-details**?nf-instance-id=<NF Instance ID>**
nf-fqdn	string	NF Profile FQDN	{apiRoot}/nrf-state-data/v1/nf-details**?nf-fqdn=<Profile level NF FQDN>**
nf-status	string	NF Profile Status	{apiRoot}/nrf-state-data/v1/nf-details**?nf-status=<Profile level NF Status>**

Query result parameters supported by nf-details URI

Note:

- These attributes can be used to get specific attributes of NF Profile in the response. User can mention them as wish to see in the response.
- These attributes can be mention in query using **result-attributes=**<Requested Attribute 1>,<Requested Attribute 2>
- Different query result attributes can be mention together comma (,) separated in query to get specific results.
- In case no result attribute is mentioned, then only **nf-instance-id** and **nf-fqdn** will be returned.
- Some of the attributes are optional in NFProfile. In case specific result attribute is asked but it is not present in NFProfile then its value will be marked UNKNOWN.

Table 7-3 Query result attributes supported by nf-details URI

Name	Details	Query Example with API
fqdn	NF Profile FQDN required in Result attributes	{apiRoot}/nrf-state-data/v1/nf-details**?nf-instance-id=<NF Instance ID>&result-attributes=fqdn**
nfType	NF Type required in Result attributes	{apiRoot}/nrf-state-data/v1/nf-details**?nf-instance-id=<NF Instance ID>&result-attributes=nftype**
nfServices	NF Services (ServiceInstanceId and Service Name) required in Result attributes	{apiRoot}/nrf-state-data/v1/nf-details**?nf-instance-id=<NF Instance ID>&result-attributes=nfServices**
nfStatus	NF Profile Status required in Result attributes	{apiRoot}/nrf-state-data/v1/nf-details**?nf-fqdn=<NF FQDN>&result-attributes=nfStatus**

Response structure supported by nf-details URI

Table 7-4 NFProfileDetails

Name	Details	Response Example

dataTimeStamp	Timestamp when data was returned	<pre>{
		
		"dataTimeStamp":"2020-11-24T15:55:
		48.000Z",
		"nfProfileDataCount":3,
		"nfProfileData":
		[{"nfInstanceId":"13515195-
		c537-4645-9b97-96ec797fbbbe","fqdn
		":"ocamf1.oracle.com"},
		
		{"nfInstanceId":"23515195-
		c537-4645-9b97-96ec797fbbbe","fqdn
		":"ocpcf1.oracle.com"},
		
		{"nfInstanceId":"33515195-
		c537-4645-9b97-96ec797fbbbe","fqdn
		":"ocudr1.oracle.com"}
		]
		
		}</pre>

nfProfileDataCount	Count of NF profile data elements in response	same as above
nfProfileData	NF Profile data attributes requested in Query result attributes	same as above

Details of subscription-details URI

Query request parameters supported by subscription-details URI

Note:

- These attributes can be used to get results based on specific input attributes. These attribute are optional to API. In case no query request and result attribute is mentioned, then only subscription ids will be provided for all of the Subscriptions.
- In case complete subscription is needed, query request attributes shall contain specific attributes **subscription-id**. Otherwise only Subscription Ids will be provided by default and additionally query result attributes can be provided to get additional details.
- At-most one query request attribute is supported, data is returned based on the request attribute.

Table 7-5 Query request attributes supported by subscription-details URI

Name	Data Type	Details	Query Example with API
subscription-id	string	Subscription Id for which data is required	{apiRoot}/nrf-state-data/v1/subscription-details**?subscription-id=<SUBSCRIPTION ID>**
nf-status-notification-uri	string	NF Status Notification URI for which data is required	{apiRoot}/nrf-state-data/v1/subscription-details?**nf-status-notification-uri=<NF Status Notification URI>**

Query result parameters supported by subscription-details URI

Note:

- These attributes can be used to get specific attributes of Subscription in the response. User can mention them as wish to see in the response.
- These attributes can be mention in query using **result-attributes=**<Requested Attribute 1>,<Requested Attribute 2>
- Different query result attributes can be mention together comma (,) separated in query to get specific results.
- In case no result attribute is mentioned, then only default attributes (subscriptionId) will be returned.
- Some of the above attributes are Optional in Subscription data. In case specific result attribute is asked but it is not present in Subscription then its value will be marked UNKNOWN.

Table 7-6 Query result attributes supported by subscription-details URI

Name	Details	Query Example with API
reqNfFqdn	Requestor NF FQDN required in Result attributes	{apiRoot}/nrf-state-data/v1/ subscription-details?**result-attributes=reqNfFqdn,reqNfType**
reqNfType	Requestor NF Type required in Result attributes	{apiRoot}/nrf-state-data/v1/subscription-details?**subscription-id=<SUBSCRIPTION ID>&result-attributes=reqNfType**
nfStatusNotificationUri	NF Status Notification URI required in Result attributes	{apiRoot}/nrf-state-data/v1/subscription-details?**subscription-id=<SUBSCRIPTION ID>&result-attributes=reqNfType**
validityTime	Validity Time required in Result attributes	{apiRoot}/nrf-state-data/v1/subscription-details?**subscription-id=<SUBSCRIPTION ID>&result-attributes=reqNfType**

Response structure supported by subscription-details URI

Table 7-7 SubscriptionDetails

Name	Details	Response Example

dataTimeStamp	Timestamp when data was returned	<pre>{
		"dataTimeStamp":
		"2020-11-24T15:55:48.000Z",
		"subscriptionDataCount": 3,
		"subscriptionData":
		[{"subscriptionId":"a1c5600116e9403bb032b214d564b729","reqNfFqdn":"amf1.oracle.com","reqNfType":"AMF"},
		{"subscriptionId":"b1c5600116e9403bb032b214d564b729","reqNfFqdn":"pcf1.oracle.com","reqNfType":"PCF"},
		{"subscriptionId":"c1c5600116e9403bb032b214d564b729","reqNfFqdn":"amf2.oracle.com","reqNfType":"UNKNOWN"}
		]
		}</pre>

subscriptionDataCount	Count of subscription data elements in response	same as above
subscriptionDataData	Subscription data attributes requested in query result attributes	same as above

Refer to OCNRF Alerts and OCNRF KPIs sections alerts and KPIs for more information respectively.

Sample Queries

Following are the queries with examples.

Query#1 - Fetches all of the NF InstanceIds

Sample query: `{apiRoot}/nrf-state-data/v1/nf-details`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
    "nfProfileDataCount":3,
    "nfProfileData":[{"nfInstanceId":"13515195-
c537-4645-9b97-96ec797fbbbe","fqdn":"ocamf1.oracle.com"},
                     {"nfInstanceId":"23515195-
c537-4645-9b97-96ec797fbbbe","fqdn":"ocpcf1.oracle.com"},
                     {"nfInstanceId":"33515195-
c537-4645-9b97-96ec797fbbbe","fqdn":"ocudr1.oracle.com"}
                    ]

}
```

Query#2 - Fetches all of the NF InstanceIds along with requested additional result attributes
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
    "nfProfileDataCount":3,
    "nfProfileData":[{"nfInstanceId":"13515195-
c537-4645-9b97-96ec797fbbbe","fqdn":"ocpcf1.oracle.com","nfType":"PCF"},

                     {"nfInstanceId":"23515195-
c537-4645-9b97-96ec797fbbbe","fqdn":"ocpcf2.oracle.com","nfType":"PCF"},

                     {"nfInstanceId":"33515195-
c537-4645-9b97-96ec797fbbbe","fqdn":"ocudr1.oracle.com","nfType":"UDR"}
                    ]
}
```

Query#3 - Fetches complete NF Profile based on NF Instance ID
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details?nf-instance-id=<NF Instance ID>`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
    "nfProfileDataCount":1,
```

```
    "nfProfileData":[{<< !!!!Complete NF Profile!!!! >>}]
}
```

Query#4 - Fetches NF Profile FQDN attribute value based on NF Instance ID
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details?nf-instance-id=<NF Instance ID>&result-attributes=fqdn`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
    "nfProfileDataCount":1,
    "nfProfileData":[{"nfInstanceId":"33515195-
c537-4645-9b97-96ec797fbbbe","fqdn":"ocpcf1.oracle.com"}]
}
```

Query#5 - Fetches NF Services attribute of NF Profile based on NF Instance ID
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details?nf-instance-id=<NF Instance ID>&result-attributes=nfServices`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
    "nfProfileDataCount":1,
    "nfProfileData":[{"nfInstanceId":"33515195-
c537-4645-9b97-96ec797fbbbe","nfServices":
[{"<serviceInstanceId":"aaaa","serviceName":"ABC"},

                  {"<serviceInstanceId":"bbbb","serviceName":"XYZ"}]
                     }]
 }
```

Query#6 - Fetches complete NF Profile based on NF FQDN
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details?nf-fqdn=<NF FQDN>`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
    "nfProfileDataCount":1,
    "nfProfileData":[{<< !!!!Complete NF Profile!!!! >>}]
}
```

Query#7 - Fetches NF Status attribute value of NF Profile based on NF Instance ID
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details?nf-instance-id=<NF Instance ID>&result-attributes=nfStatus`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
```

```
    "nfProfileDataCount":1,
    "nfProfileData":[{"nfInstanceId":"33515195-
c537-4645-9b97-96ec797fbbbe","nfStatus":"Suspended"}]
}
```

Query#8 - Fetches NFInstance Id and its NF Status attribute value based on NF FQDN
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details?nf-fqdn=<NF FQDN>&result-attributes=nfStatus`

Sample response:

```
{
    "dataTimeStamp":"2020-11-24T15:55:48.000Z",
    "count":3,
    "nfProfileData":[{"nfInstanceId":"13515195-
c537-4645-9b97-96ec797fbbbe","nfStatus":"Suspended"}]
}
```

Query#9 - Fetches NF Profile details based on NF Status
Sample query: `{apiRoot}/nrf-state-data/v1/nf-details?nf-status=<NF Status>&result-attributes=fqdn,nfType`

Sample response:

```
{
    "dataTimeStamp": "2020-11-24T15:55:48.000Z",
    "nfProfileDataCount": 3,
    "nfProfileData": [{"nfInstanceId":"13515195-
c537-4645-9b97-96ec797fbbbe", "fqdn":"ocpcf1.oracle.com",
"nfType":"PCF"},
                      {"nfInstanceId":"23515195-
c537-4645-9b97-96ec797fbbbe", "fqdn":"ocpcf2.oracle.com",
"nfType":"PCF"},
                      {"nfInstanceId":"33515195-
c537-4645-9b97-96ec797fbbbe", "fqdn":"ocudr1.oracle.com",
"nfType":"UDR"}
                    ]
}
```

Query#10 - Fetches all of the subscriptions
Sample query: `{apiRoot}/nrf-state-data/v1/subscription-details`

Sample response:

```
{
    "dataTimeStamp": "2020-11-24T15:55:48.000Z",
    "subscriptionDataCount": 3,
    "subscriptionData": [{"subscriptionId":
"a1c5600116e9403bb032b214d564b729"},
                         {"subscriptionId":
"b1c5600116e9403bb032b214d564b729"},
                         {"subscriptionId":
"c1c5600116e9403bb032b214d564b729"}
```

```
                        ]
}
```

Query#11- Fetches all of the subscriptions along with requested result attributes
Sample query: `{apiRoot}/nrf-state-data/v1/subscription-details?result-attributes=reqNfFqdn,reqNfType`

Sample response:

```
{
    "dataTimeStamp": "2020-11-24T15:55:48.000Z",
    "subscriptionDataCount": 3,
    "subscriptionData":
[{"subscriptionId":"a1c5600116e9403bb032b214d564b729","reqNfFqdn":"amf1.
oracle.com","reqNfType":"AMF"},

{"subscriptionId":"b1c5600116e9403bb032b214d564b729","reqNfFqdn":"pcf1.o
racle.com","reqNfType":"PCF"},

{"subscriptionId":"c1c5600116e9403bb032b214d564b729","reqNfFqdn":"amf2.o
racle.com","reqNfType":"UNKNOWN"}
                        ]
}
```

Query#12 - Fetches Subscription Data based on Subscription ID
Sample query: `{apiRoot}/nrf-state-data/v1/subscription-details?subscription-id=<SUBSCRIPTION ID>`

Sample response:

```
{
  "dataTimeStamp":"2020-11-24T15:55:48.000Z",
  "subscriptionDataCount":3,
  "subscriptionData":[<< !!!!Complete Subscription Data!!!!>>]
}
```

Query#13 - Fetches specific attributes of Subscription Data based on Subscription ID
Sample query: `{apiRoot}/nrf-state-data/v1/subscription-details?subscription-id=<SUBSCRIPTION ID>&result-attributes=reqNfFqdn,reqNfType`

Sample response:

```
{
    "dataTimeStamp": "2020-11-24T15:55:48.000Z",
    "subscriptionDataCount": 3,
    "subscriptionData":
[{"subscriptionId":"a1c5600116e9403bb032b214d564b729","reqNfFqdn":"amf1.
oracle.com","reqNfType":"AMF"},

{"subscriptionId":"b1c5600116e9403bb032b214d564b729","reqNfFqdn":"pcf1.o
racle.com","reqNfType":"PCF"},

{"subscriptionId":"c1c5600116e9403bb032b214d564b729","reqNfFqdn":"amf2.o
```

```
racle.com","reqNfType":"UNKNOWN"}
                        ]
}
```

Query#14 - Fetches Subscription IDs based on nf-status-notification-uri
Sample query: `{apiRoot}/nrf-state-data/v1/subscription-details?nf-status-notification-uri=<NF Status Notification URI>&result-attributes=reqNfFqdn,reqNfType`

Sample response:

```
{
    "dataTimeStamp": "2020-11-24T15:55:48.000Z",
    "subscriptionDataCount": 2,
    "subscriptionData":
[{"subscriptionId":"a1c5600116e9403bb032b214d564b729","reqNfFqdn":"amf1.
oracle.com","reqNfType":"AMF"},

{"subscriptionId":"b1c5600116e9403bb032b214d564b729","reqNfFqdn":"UNKNOW
N","reqNfType":"PCF"}
                        ]
}
```

8
OCNRF configuration status REST APIs

API details

apiRoot is concatenation of the following parts:

- scheme:- http, https
- the fixed string "://"
- authority (host and optional port) host and port will be CNCC host and port details

Table 8-1 API Details

API	HTTP method supported	Description	HTTP response codes
{apiRoot}/nrf-status-data/v1/accessTokenSigningDataStatus	GET	This API fetches Access Token Signing Data Status from OCNRF. OCNRF provides option to configure access token signing key and certificate details. Using this API, it can be checked that details provided are valid or not and specific key details can be used to sign the token.	200 OK with **AccessTokenSigningDataStatus**, if Access Token Signing data details found. 200 OK with Empty List <**AccessTokenSigningData**> inside **AccessTokenSigningDataStatus**, if Access Token Signing data details not found.

API example

Sample API:- *{apiRoot}/nrf-status-data/v1/accessTokenSigningDataStatus*

```
Method:- GET
Sample response:-
{
"dataTimeStamp":"2020-11-24T15:55:48.000Z",
"accessTokenSigningKeysCount": 2,
"accessTokenSigningData":[{"keyID": "KeyId01", "privateKey":
{ "fileName": "KeyId01-privateKey.pem", "isValid": true,
"invalidReason": null },
                                                "certificate":
{ "fileName": "KeyId01-publicCertificate.crt", "isValid": true,
"invalidReason": null, "expiryTime": "2021-11-24T15:55:48.000Z"}
                          },
                          {"keyID": "KeyId02", "privateKey":
{ "fileName": "KeyId02-privateKey.pem", "isValid": false,
"invalidReason": "Key file not found" },
                                                "certificate":
{ "fileName": "KeyId02-publicCertificate.crt", "isValid": false,
"invalidReason": "Key file not found", "expiryTime":null }
```

```
                        },
                        ]
}
```

Data Models

Table 8-2 AccessTokenSigningDataStatus

Attribute	DataType	Description
dataTimeStamp	string	Time stamp when Data was retrieved
accessTokenSigningKeysCount	integer	Count of keys in response
accessTokenSigningData	array(AccessTokenSigningData)	See AccessTokenSigningData for details

Table 8-3 AccessTokenSigningData

Attribute	DataType	Description
keyID	string	Key Id for the Access Token Signing Data
privateKey	AccessTokenSigningDataDetails	Private key details corresponding to KeyId
certificate	AccessTokenSigningDataDetails	Public Certificate details corresponding to KeyId

Table 8-4 AccessTokenSigningDataDetails

Attribute	DataType	Description
fileName	string	File Name of the Private Key and Public Certificate
isValid	boolean (true or false)	Details provided are valid to use or not.
invalidReason	string	In case `isValid` value is false then the reason why the key or certificate is not valid is provided are `invalidReason`.
expiryTime	string	This attribute is applicable only for certificate. This will tell when certificate will get expire.

9
Configuring OCNRF using CNC Console

This section provides information for configuring Oracle Communications Network Repository Function.

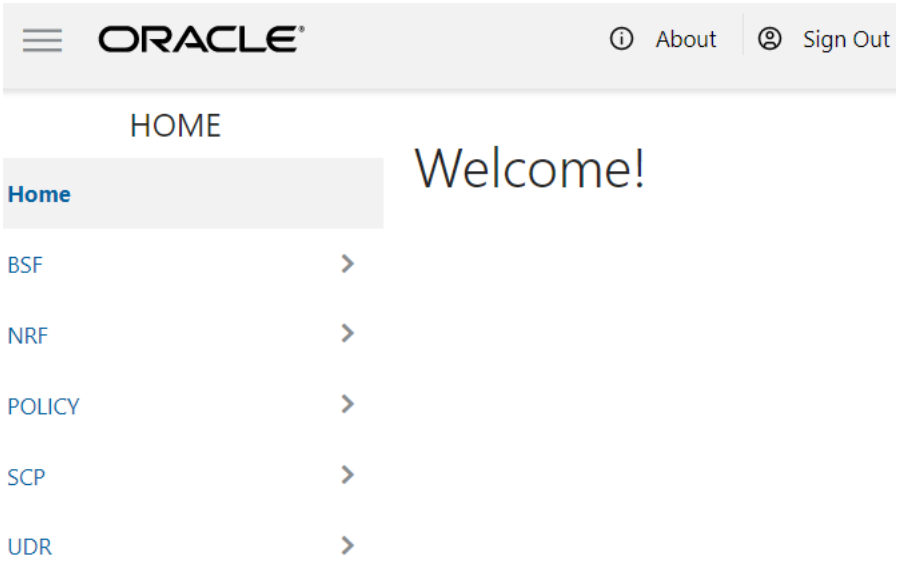
CNC Console Interface

CNC Console Login

Following is the procedure to login to CNC Console:

1. Open any browser.
2. Enter the URL: ***http://<host name>:<port number>***.
3. Enter valid credentials.
4. Click **Log in**. The CNC Console interface is displayed.

Figure 9-1 CNC Console



Top Ribbon

The top ribbon has following options:

1. About
2. Sign Out
3. Help

> **Note:**
>
> The Collapse button at the left side allows the user to collapse the left pane. Help navigates to the swagger.

Left Pane - NFs and APIs

The left pane displays the list of Network Functions and respective APIs.

Right Pane - Details View

The right pane displays details of the parameters that can be updated in the selected API.

OCNRF Configuration

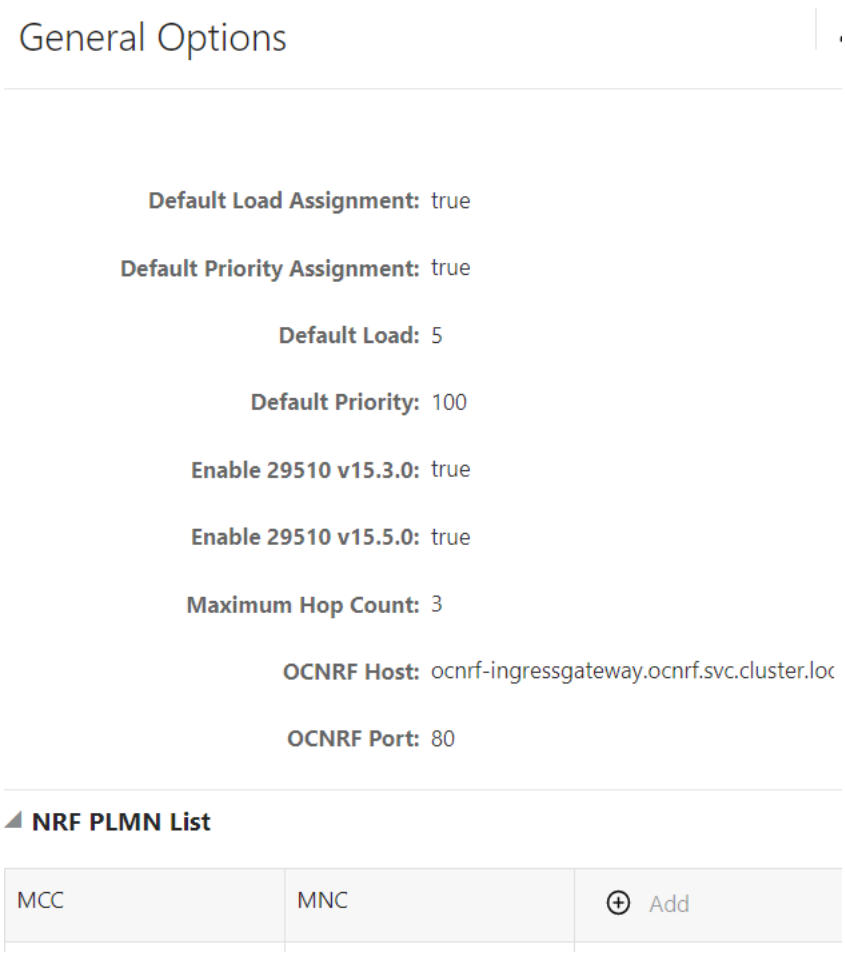
This section provides configuration steps for OCNRF parameters using CNC Console.

General Options

To configure the general options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **General Options**. The **General Options** screen is displayed.

Figure 9-2 General Options



2. Click **Edit** from the top right side to edit or update **General Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in General Options.
4. Click **Add** provided under **NRF PLMN List** section to add MCC and MNC.
5. Add **MCC** and **MNC** values in the field.
6. Click **Save** on **Add NRF PLMN List** window.
7. Click **Save** to save the general options.

Editing NRF PLMN List

To edit an existing NRF PLMN List:

1. Click **Edit** from the top right side to edit or update **General Options** parameter. The screen is enabled for modification.
2. Click **Edit** provided under **NRF PLMN List** section to edit the PLMN list.
3. Modify the attribute values as per the requirement.
4. Click **Save** on **Edit NRF PLMN List** window.
5. Click **Save** to save the updated general options.

Deleting NRF PLMN List

To delete a NRF PLMN List Parameter:

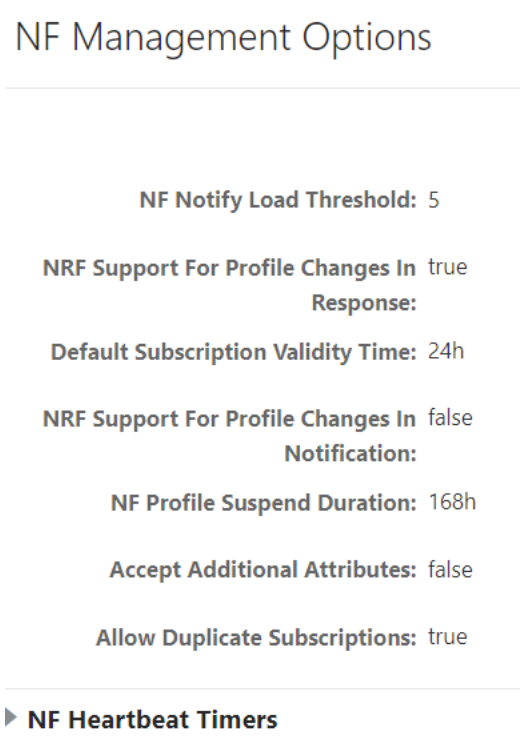
1. Click **Edit** from the top right side to edit or update **General Options** parameter. The screen is enabled for modification.
2. Click **Delete** from the action items of **NRF PLMN List** section under each *Rules Data*.
   The *" Do you want to delete the record?"* message appears.
3. Click **OK** to delete the parameter.
4. Click **Save** to save the updated general options.

NF Management Options

To configure the NF Management Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **NF Management Options**. The **NF Management Options** screen is displayed.

Figure 9-3 NF Management Options



2. Click **Edit** from the top right side to edit or update **NF Management Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in NF Management Options.
4. Click **Add** provided under **NF Heartbeat Timers** section to add the Heartbeat Timers parameters.
5. Enter the values for heartbeat timers as described in NF Management Options.

6. Click **Save** on **Add NF Heartbeat Timers** window.
7. Click **Save** to save the NF Management Options.

Editing NF Heartbeat Timers

To edit an existing NF Heartbeat Timers:

1. Click **Edit** from the top right side to edit or update **NF Management Options** parameter. The screen is enabled for modification.
2. Click **Edit** provided under **NF Heartbeat Timers** section to edit the host information.
3. Modify the attribute values as per the requirement.
4. Click **Save** on **Edit NF Heartbeat Timers** window.
5. Click **Save** to save the updated NF Management Options.

Deleting NF Heartbeat Timers

To delete a NF Heartbeat Timers Parameter:

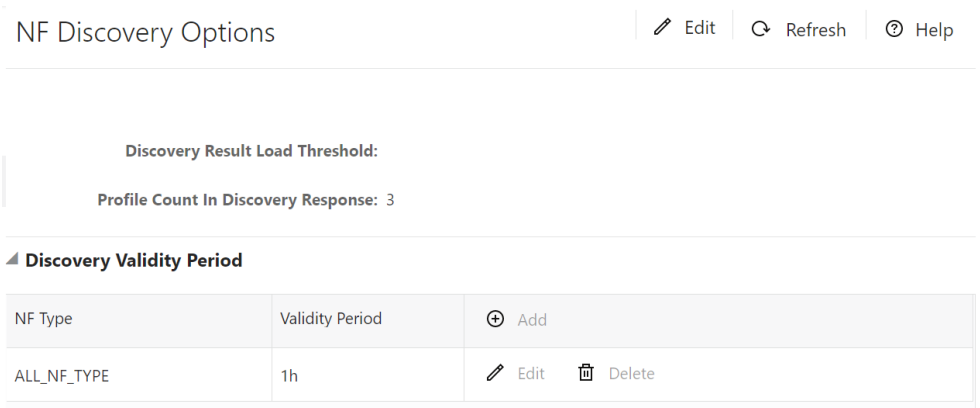
1. Click **Edit** from the top right side to edit or update **NF Management Options** parameter. The screen is enabled for modification.
2. Click **Delete** from the action items of **Edit NF Heartbeat Timers** section under each *NF Heartbeat Timers*.
   The *" Do you want to delete the record?"* message appears.
3. Click **OK** to delete the parameter.
4. Click **Save** to save the updated NF Management Options.

NF Discovery Options

To configure the NF Discovery Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **NF Discovery Options**. The **NF Discovery Options** screen is displayed.

Figure 9-4 NF Discovery Options



2. Click **Edit** from the top right side to edit or update **NF Discovery Options** parameter. The screen is enabled for modification.

3. Provide the values for the attributes as described in NF Discovery Options:
   a. Increment or decrement the **Discovery Result Load Threshold** parameter using the arrow keys.
   b. Increment or decrement the **Profiles Count In Discovery Response** parameter using the arrow keys.
4. Click **Add** provided under **Discovery Validity Period** section to add the validity information.
5. Enter the values for rules configuration as described in NF Discovery Options.
6. Click **Save** on **Add Discovery Validity Period** window.
7. Click **Save** to save the NF Discovery Options.

Editing Discovery Validity Period

To edit an existing Discovery Validity Period:

1. Click **Edit** from the top right side to edit or update **NF Discovery Options** parameter. The screen is enabled for modification.
2. Click **Edit** provided under **Discovery Validity Period** section to edit the host information.
3. Modify the attribute values as per the requirement.
4. Click **Save** on **Edit Discovery Validity Period** window.
5. Click **Save** to save the updated NF Discovery Options.

Deleting NF Type

To delete a NF type:

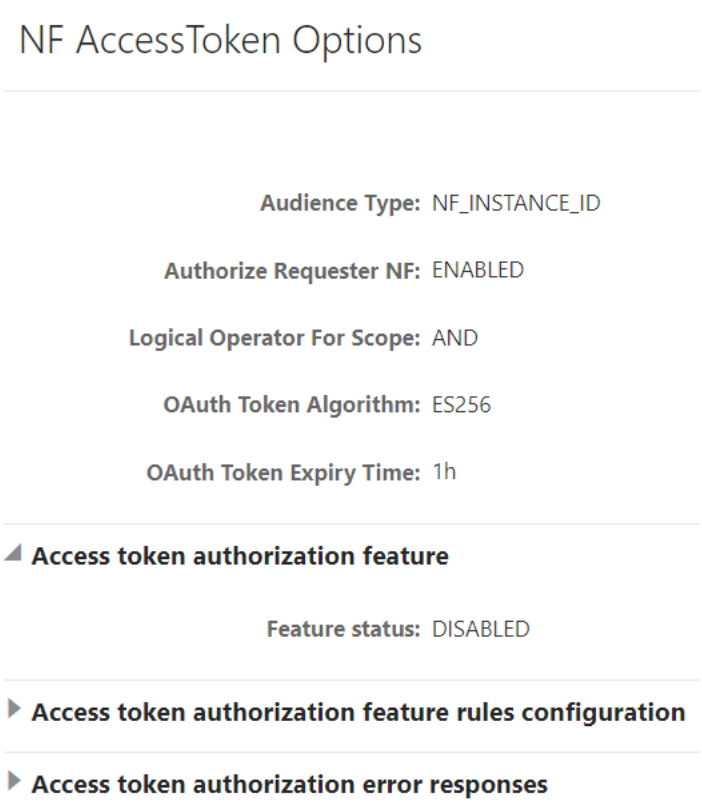
1. Click **Edit** from the top right side to edit or update **NF Discovery Options** parameter. The screen is enabled for modification.
2. Click **Delete** from the action items of **Discovery Validity Period** section. The *" Do you want to delete the record?"* message appears.
3. Click **OK** to delete the NF type details.
4. Click **Save** to save the updated NF Discovery Options.

NF Access Token Options

To configure the NF Access Token Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **NF Access Token Options**. The **NF Access Token Options** screen is displayed.

Figure 9-5 NF Access Token Options



2. Click **Edit** from the top right side to edit or update **NF Access Token Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in NF Access Token Options.
4. Click **Add** provided under **Access token authorization feature rules configuration** section to add the rules.
5. Enter the values for rules configuration as described in NF Access Token Options.
6. Click **Save** on **Add Access token authorization error responses** window.
7. Click **Add** provided under **Access token authorization error responses** section to add the error response details.
8. Enter the values for rules configuration as described in NF Access Token Options.
9. Click **Save** on **Add Access token authorization error responses** window.
10. Click **Save** to save the NF Access Token Options.

Editing NF Access Token Options

To edit an existing NF Access Token Options:

1. Click **Edit** from the top right side to edit or update **NF Access Token Options** parameter. The screen is enabled for modification.
2. Modify the attribute values as per the requirement.
3. Click **Save** on **Edit NF Access Token Options** window.
4. Click **Save** to save the updated NF Access Token Options.

Deleting NF Access Token Options

To delete a NF Access Token Options:

1. Click **Edit** from the top right side to edit or update **NF Access Token Options** parameter. The screen is enabled for modification.
2. Click **Delete** from the action items of **Edit NF Access Token Options** section under each *Access token authorization feature rules configuration*.
   The *" Do you want to delete the record?"* message appears.
3. Click **OK** to delete the parameter.
4. Click **Save** to save the updated NF Access Token Options.

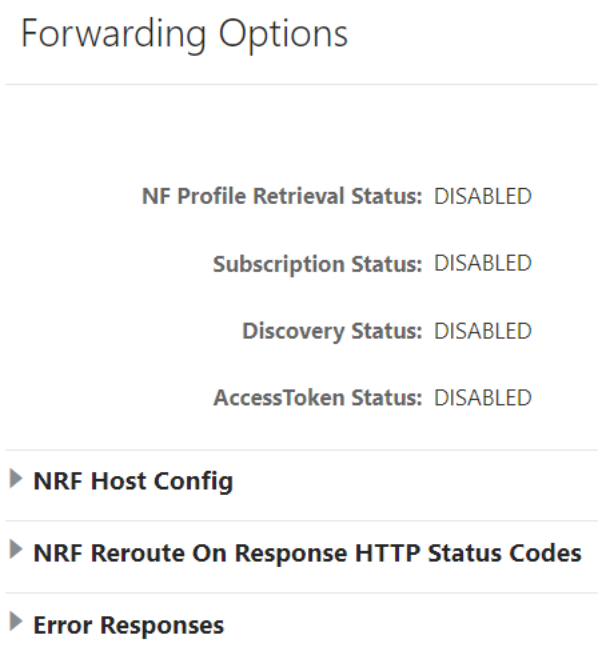
> **Note:**
>
> Follow above procedures for editing or deleting **Access token authorization error responses**.

Forwarding Options

To configure the Forwarding Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **Forwarding Options**.
   The **Forwarding Options** screen is displayed.

Figure 9-6 Forwarding Options



2. Click **Edit** from the top right side to edit or update **Forwarding Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in NRF-NRF Forwarding Options.

4. Click **Add** provided under **NRF Host Config** section to add the host information.
5. Enter the values for rules configuration as described in NRF-NRF Forwarding Options.
6. Click **Save** on **Add NRF Host Config** window.
7. Click **Add** provided under **Error Responses** section to add the error response details.
8. Enter the values for rules configuration as described in NRF-NRF Forwarding Options.
9. Click **Save** on **Add Error Responses** window.
10. Click **Save** to save the Forwarding Options.

Editing NRF Host Config

To edit an existing NRF Host Config:

1. Click **Edit** from the top right side to edit or update **Forwarding Options** parameter. The screen is enabled for modification.
2. Click **Edit** provided under **NRF Host Config** section to edit the host information.
3. Modify the attribute values as per the requirement.
4. Click **Save** on **Edit NRF Host Config** window.
5. Click **Save** to save the updated Forwarding Options.

Deleting NRF Host Config

To delete a NRF Host Config:

1. Click **Edit** from the top right side to edit or update **Forwarding Options** parameter. The screen is enabled for modification.
2. Click **Delete** from the action items of **NRF Host Config** section.
   The *" Do you want to delete the record?"* message appears.
3. Click **OK** to delete the host config details.
4. Click **Save** to save the updated Forwarding Options.

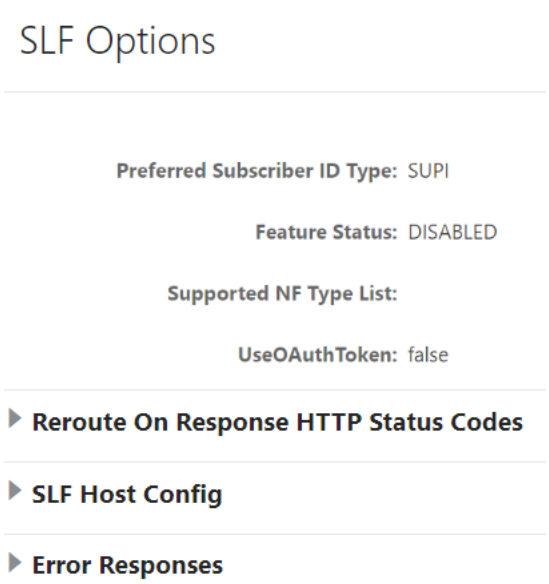
> **Note:**
>
> Follow above procedures for editing or deleting the **Error Responses**.

SLF Options

To configure the SLF Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **SLF Options**. The **SLF Options** screen is displayed.

Figure 9-7 SLF Options



2. Click **Edit** from the top right side to edit or update **SLF Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in SLF Options.
4. In **Edit** mode, **Add** HTTP status code information under **Reroute On Response HTTP Status Codes** section.
5. Enter the values for HTTP status codes as described in SLF Options.
6. Click **Add** provided under **SLF Host Config** section to add the host information.
7. Enter the values for rules configuration as described in SLF Options.
8. Click **Save** on **Add SLF Host Config** window.
9. Click **Add** provided under **Error Responses** section to add the error response details.
10. Enter the values for rules configuration as described in SLF Options.
11. Click **Save** on **Add Error Responses** window.
12. Click **Save** to save the SLF Options.

Editing SLF Host Config

To edit an existing SLF Host Config:

1. Click **Edit** from the top right side to edit or update **SLF Options** parameter. The screen is enabled for modification.
2. Click **Edit** provided under **SLF Host Config** section to edit the host information.
3. Modify the attribute values as per the requirement.
4. Click **Save** on **Edit SLF Host Config** window.
5. Click **Save** to save the updated SLF Options.

Deleting SLF Host Config

To delete a SLF Host Config:

1. Click **Edit** from the top right side to edit or update **SLF Options** parameter. The screen is enabled for modification.
2. Click **Delete** from the action items of **SLF Host Config** section.
   The *" Do you want to delete the record?"* message appears.
3. Click **OK** to delete the host config details.
4. Click **Save** to save the updated SLF Options.

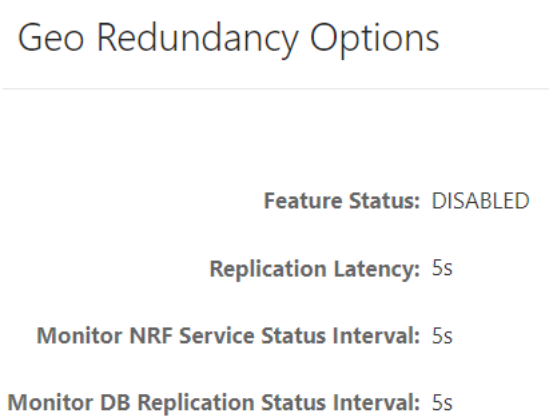
> **Note:**
>
> Follow above procedures for editing or deleting the **Error Responses**.

Geo Redundancy Options

To configure the Geo Redundancy Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **Geo Redundancy Options**. The **Geo Redundancy Options** screen is displayed.

Figure 9-8 Geo Redundancy Options



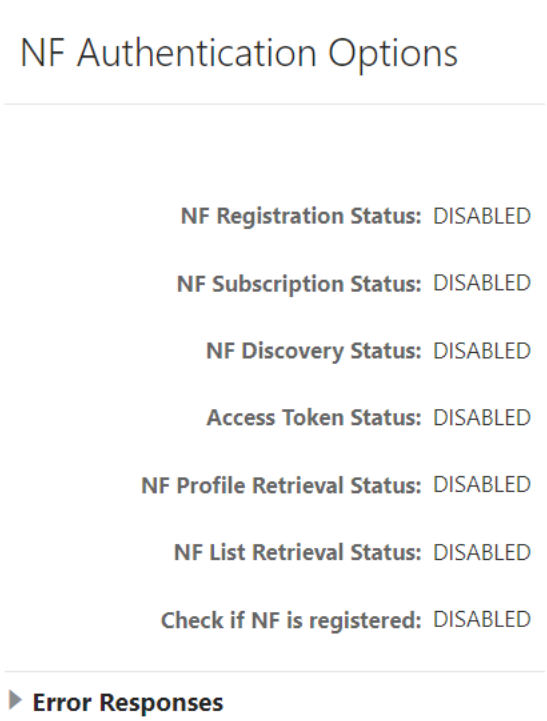
2. Click **Edit** from the top right side to edit or update **Geo Redundancy Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in Geo Redundancy Options:
   a. Select the **Feature Status** from the drop-down menu.
   b. Enter the values for **Replication Latency**, **Monitor NRF Service Status Interval** and **Monitor DB Replication Status Interval**.
4. Click **Save**.

NF Authentication Options

To configure the NF Authentication Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **NF Authentication Options**. The **NF Authentication Options** screen is displayed.

Figure 9-9 NF Authentication Options



2. Click **Edit** from the top right side to edit or update **NF Authentication Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in NF Authentication Options.
4. Click **Add** provided under **Error Responses** section to add the error response details.
5. Enter the values for rules configuration as described in NF Authentication Options.
6. Click **Save** on **Add Error Responses** window.
7. Click **Save** to save the NF Authentication Options.

Editing Error Responses

To edit an existing Error Responses:

1. Click **Edit** from the top right side to edit or update **NF Authentication Options** parameter. The screen is enabled for modification.
2. Click **Edit** provided under **Error Responses** section to edit the error response details.
3. Modify the attribute values as per the requirement.

4. Click **Save** on **Edit Error Responses** window.
5. Click **Save** to save the updated NF Authentication Options.

Deleting Error Responses

To delete a Error Responses:

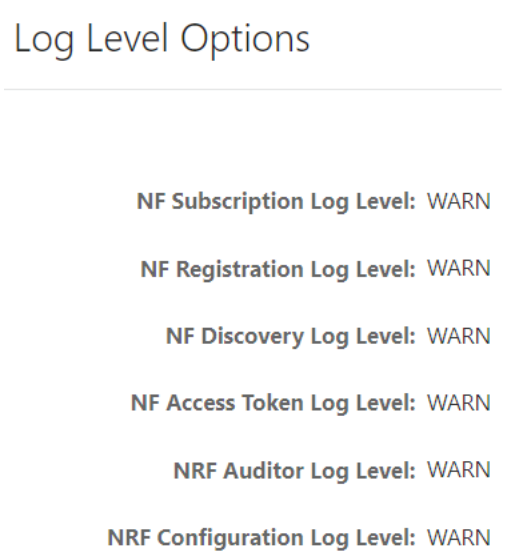
1. Click **Edit** from the top right side to edit or update **NF Authentication Options** parameter. The screen is enabled for modification.
2. Click **Delete** from the action items of **Error Responses** section.
   The *" Do you want to delete the record?"* message appears.
3. Click **OK** to delete the error response details.
4. Click **Save** to save the updated NF Authentication Options.

Log Level Options

To configure the Log Level Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **Log Level Options**. The **Log Level Options** screen is displayed.

Figure 9-10 Log Level Options



2. Click **Edit** from the top right side to edit or update **Log Level Options** parameter. The screen is enabled for modification.
3. Select the values for the following attributes from the drop-down menu:
   a. NF Subscription Log Level
   b. NF Registration Log Level
   c. NF Discovery Log Level
   d. NF Access Token Log Level
   e. NRF Auditor Log Level

f. NRF Configuration Log Level

Refer to Log Level Options for more information.

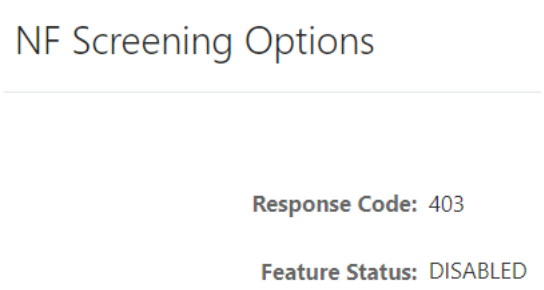
4. Click **Save**.

NF Screening Options

To configure the NF Screening Options follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **NF Screening Options**. The **NF Screening Options** screen is displayed.

Figure 9-11 NF Screening Options



2. Click **Edit** from the top right side to edit or update **NF Screening Options** parameter. The screen is enabled for modification.
3. Provide the values for the attributes as described in NF Screening Options:
   a. Enter the **Response Code** value in the field.
   b. Select the **Feature Status** from the drop-down menu.
4. Click **Save**.

Screening Rules

NF Screening supports the functionality to screen the service requests received from 5G Network Functions (NFs) before allowing access to OCNRF services. In this feature, OCNRF screens the incoming service operations from NFs on the basis of some attributes against set of rules configured at OCNRF. OCNRF processes the required services only if screening is successful. This feature provides extra security by restricting the NF that can use the service of OCNRF.

Using the screening lists, operator can decide which NF can access the services provided by OCNRF by configuring attributes based on the requirement.

CALLBACK URI

Screening list type for callback URIs in NF Service and nfStatusNotificationUri in SubscriptionData.

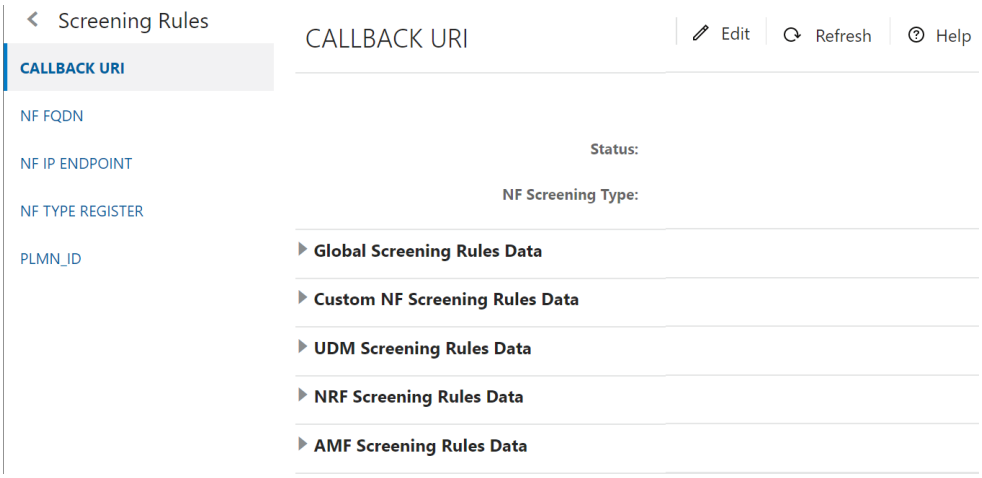
NRF screens the callback URI present in the request before allowing access to management service. Host present in callback URI (FQDN+port or IP+port) must be used for screening. In CALLBACK URI, the attributes that can be modified are FQDN, Port and IP address.

Configuring Callback URI Parameters

To configure Callback URI parameters follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **Screening Rules**. Under **Screening Rules** select **CALLBACK URI**. The **CALLBACK URI** screen is displayed.

Figure 9-12 Callback URI



2. Click **Edit** from the top right side to edit or update a CALLBACK URI parameter. The screen is enabled for modification.
3. Provide the values for the attributes as follows:
   a. Select **Status** from drop-down menu.
   b. Select the required **NF Screening Type** from drop-down menu.
   c. Choose the **Failure Action** for specific *Rules Data* from drop-down menu.
   d. Click **Add** provided under **NF Callback URIs** section based on the requirement for each *Rules Data* to add NF Callback URIs. Refer to Callback URI parameters for more information in parameter values and description.
4. Click **Save**.

Modifying NF Callback URIs

The user can edit and delete the NF Callback URIs.

Editing NF Callback URIs

To edit an existing NF Callback URIs:

1. In the **Edit** Mode of **Callback URI** screen, click **Edit** from the **NF Callback URIs** section under each *Rules Data*. The **Edit NF Callback URIs** Screen appears.
2. Modify the attribute values as per the requirement.
3. Click **Save**.

Deleting NF Callback URIs

To delete a NF Callback URIs Parameter:

1. Click **Delete** from the action items of **NF Callback URIs** section under each *Rules Data*.
   The *" Do you want to delete the record"* message appears.
2. Click **OK** to delete the parameter.

NF FQDN

NRF screens the Fully Qualified Domain Name (FQDN) present in the request before allowing access to management service.

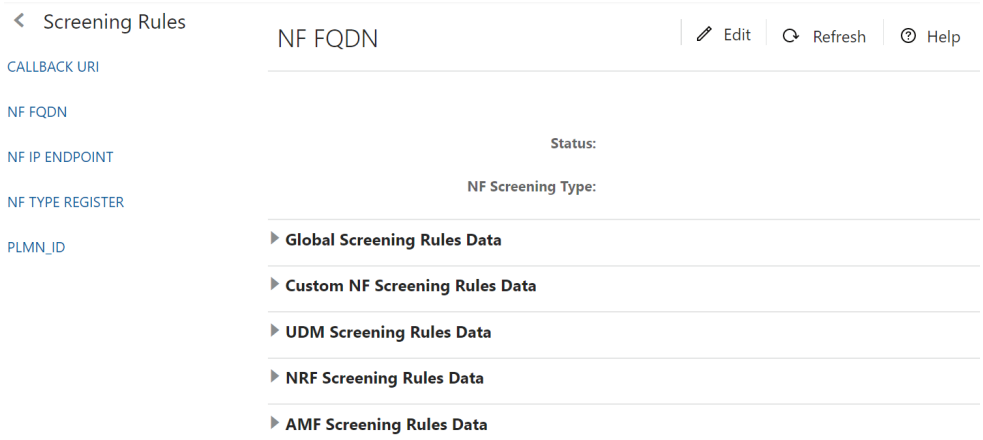
In NF FQDN, the attributes that can be modified are pattern, fqdn in NFProfile and fqdn in NFService.

Configuring NF FQDN Parameters

Follow the procedure to configure NF FQDN parameters:

1. From the left navigation menu, navigate to **NRF** and then select **Screening Rules**. Under **Screening Rules**, select **NF FQDN**. The **NF FQDN** screen is displayed.

Figure 9-13 NF FQDN



2. Click **Edit** from the top right side to edit or update the NF FQDN parameter. The **Edit NF FQDN** screen is displayed.
3. Provide the values for the attributes as follows:
   a. Select **Status** from drop-down menu.
   b. Select the required **NF Screening Type** from drop-down menu.
   c. Choose the **Failure Action** for specific *Rules Data* from drop-down menu.
   d. Enter the **Pattern** and **FQDN** values under **NF FQDN** section based on the requirement for each *Rules Data*.
4. Click **Save**.

> **Note:**
>
> Repeat the above steps for all the **Rules Data**.

NF IP Endpoint

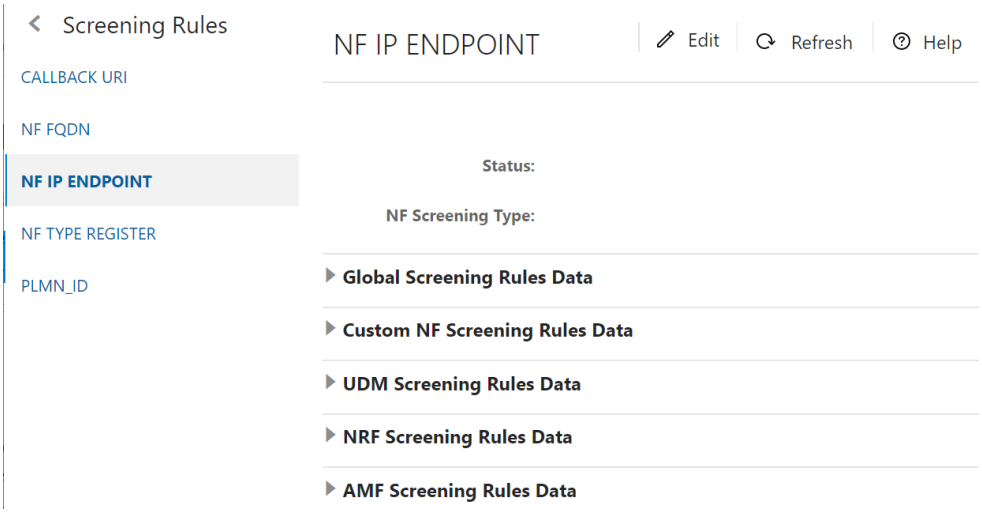
NRF screens the IP endpoint(s) present in the request before allowing access to management service.

Configuring NF IP Endpoint parameters

To configure NF IP Endpoint parameters follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **Screening Rules**. Under **Screening Rules** select **NF IP Endpoint**. The **NF IP Endpoint** screen is displayed.

Figure 9-14 NF IP Endpoints



2. Click **Edit** from the top right side to edit or update NF IP Endpoint parameters. The screen is enabled for modification.
3. Provide the values for the attributes as follows:
   a. Select **Status** from drop-down menu.
   b. Select the required **NF Screening Type** from drop-down menu.
   c. Choose the **Failure Action** for specific *Rules Data* from drop-down menu.
   d. Click **Add** provided under **NF IP Endpoint** section based on the requirement for each *Rules Data* to add NF IP Endpoint. Refer to NF IP Endpoint for more information in parameter values and description.
4. Click **Save**.

Modifying NF IP Endpoint

The user can edit or delete the NF IP Endpoint.

Editing NF IP Endpoint

To edit an existing NF IP Endpoint:

1. In the **Edit** Mode of **NF IP Endpoint** screen, click **Edit** from the **NF IP Endpoint** section under each *Rules Data*. The **Edit NF IP Endpoint** Screen appears.
2. Modify the attribute values as per the requirement.
3. Click **Save**.

Deleting NF IP Endpoint

To delete a NF IP Endpoint Parameters:

1. Click **Delete** from the action items of the **NF IP Endpoint** section under each *Rules Data*.
   The *" Do you want to delete the record"* message appears.
2. Click **OK** to delete the parameter.

NF Type Register

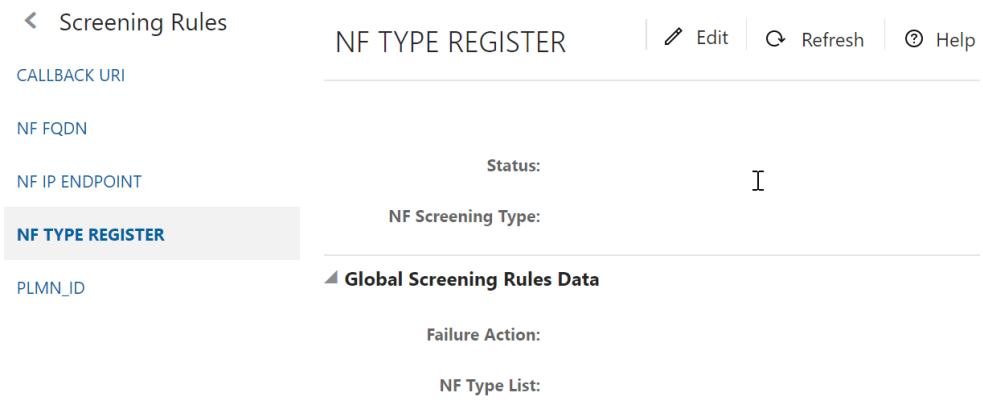
NRF screens the NF type present in the in-coming service request.

Configuring NF IP Type Register parameters

Following is the procedure to configure NF IP Type Register parameters:

1. From the left navigation menu, navigate to **NRF** and then **Screening Rules**. Under **Screening Rules** select **NF IP Type Register**. The **NF IP Type Register** screen is displayed.

Figure 9-15 NF IP Type Register



2. Click **Edit** from the top right side to edit or update a NF IP Type Register parameters. The screen is enabled for modification.
3. Provide the values for the attributes as follows:
   a. Select **Status** from drop-down menu.
   b. Select the required **NF Screening Type** from drop-down menu.
   c. Choose the **Failure Action** and **NF Type List** for *Global Screening Rules Data*.

4. Click **Save**.

PLMN ID Parameters

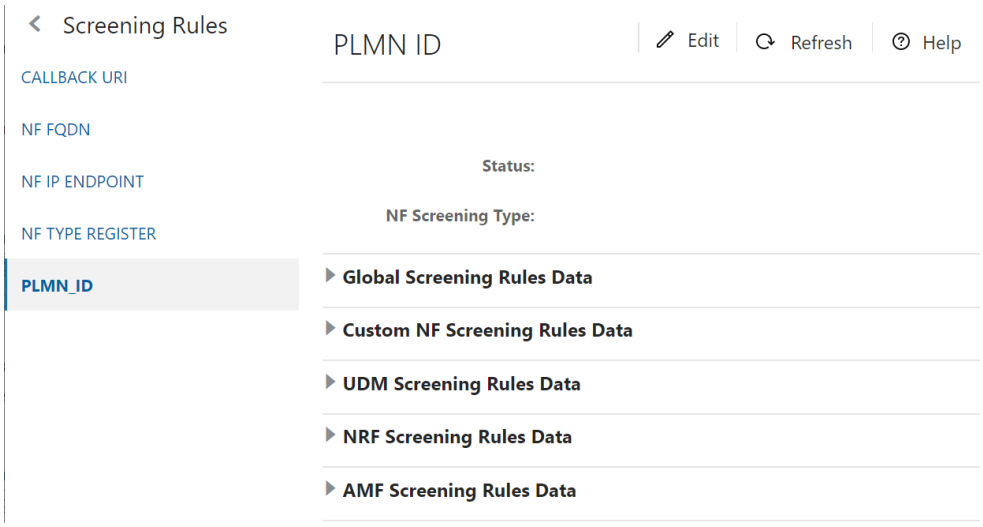
NRF screens the PLMN Id present in the request before allowing access to management service.

Configuring PLMN ID Parameters

To configure PLMN ID parameters follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **Screening Rules**. Under **Screening Rules** select **PLMN ID**. The **PLMN ID** screen is displayed.

Figure 9-16 PLMN ID



2. Click **Edit** from the top right side to edit or update a PLMN ID parameters. The screen is enabled for modification.
3. Provide the values for the attributes as follows:
   a. Select **Status** from drop-down menu.
   b. Select the required **NF Screening Type** from drop-down menu.
   c. Choose the **Failure Action** for specific *Rules Data* from drop-down menu.
   d. Click **Add** provided under **PLMN List** section based on the requirement for each *Rules Data* to add PLMN List. Refer to Configuring NF Screening for more information in parameter values and description.
4. Click **Save**.

Modifying PLMN ID

The user can edit or delete the PLMN ID.

Editing PLMN ID

To edit an existing PLMN ID:

1. In the **Edit** Mode of **PLMN ID** screen, click **Edit** from the **PLMN List** section under each *Rules Data*. The **Edit PLMN List** Screen appears.
2. Modify the attribute values as per the requirement.
3. Click **Save**.

Deleting PLMN ID

To delete a PLMN ID Parameters:

1. Click **Delete** from the action items of the **PLMN List** section under each *Rules Data*.
   The *" Do you want to delete the record"* message appears.
2. Click **OK** to delete the parameter.

10
OCNRF Metrics, KPIs, and Alerts

OCNRF Metrics

This section includes information about metrics for Oracle Communications Network Repository Function.

> **Note:**
>
> Sample OCNRF dashboard for Grafana is delivered to the customer through OCNRF Custom Templates. Metrics and functions used to achieve KPI are covered in OCNRF Custom Templates. Refer to Oracle Help Center site for the information about OCNRF Custom Templates.

Dimensions Legend for the Metrics

The following table includes the details about the metrics dimensions:

Table 10-1 Dimensions Legend

Dimension	Details
application	application name, here it is ocnrf
vendor	For OCNRF, vendor is Oracle
Method	HTTP Method Name. For example: PUT, GET
Status	HTTP Status Code in response
Uri	URI defined to identify the Service Operation at Ingress Gateway
Node	Name of the kubernetes worker node on which microservice is running
NrfLevel	OCNRF Deployment Name by which OCNRF can be identified, it will be OCNRF Instance Id passed through helm
NfType	Types of Network Functions (NF)
NfInstanceId	Unique identity of the NF Instance sending request to OCNRF
HttpStatusCode	HTTP Status Code
ServiceName	Name of the service instance (for example: "nudm-sdm")
ServiceInstanceId	Unique ID of the service instance within a given NF Instance
UpdateType(Partial/ Complete)	NF Update with PUT (Complete) or PATCH (Partial) methods
OperationType	Dimension is for NFSubscribe Service operation to tell if the request is to create or update the subscription
NotificationEventType	This dimension indicates subscription request is for which event types. For example: NF_REGISTERED, NF_DEREGISTERED and NF_PROFILE_CHANGED
TargetNfType	Dimension indicates request is for which target NF type

Table 10-1 (Cont.) Dimensions Legend

Dimension	Details
RequesterNfType	Dimension indicates the NF type which originating the request. This value comes from UserAgent header. For NFDiscover Service operation it is taken from Search Query. In case no header or value, this value will be UNKNOWN in the metrics
TargetNfInstanceId	Dimension indicates the target NF Instance Id for NF Access Token
ClientNfInstanceId	Dimension indicates the client NF Instance Id for NF Access Token
RejectionReason	Dimension indicates the rejection reason for NF Access Token
SubscriptionIdType	Dimension indicates the Subscription Id type for which SLF query is received
GroupId	Dimension indicates the GroupId returned by SLF/UDR corresponding to SubscriptionId
BucketSize	Dimension indicates how many profiles are returned in the response of Discovery request. Range is not configurable. Possible values are 0-10, +Inf. According to NF profiles returned, corresponding bucket will be incremented by one. For example, if 2 profiles are returned, then bucket 2 will be incremented by one. Profiles getting returned more than 10 will fall in +Inf bucket.
DBOperation	Create,update,delete and find
TableName	OCNRF Table Name
SubscriptionStatus	Status of subscription shall be 'SUBSCRIBED', 'SUSPENDED' or 'UNSUBSCRIBED'
DbReplicationStatus	"ACTIVE" or "INACTIVE"
RemoteNrfInstanceId	Remote OCNRF Instance Id
HeartbeatTimer	The heartbeatTimer of the NfProfile. The value is considered in seconds.
TLSFqdn	FQDN received in TLS Certificate
NfFqdn	FQDN of consumer NF. This dimension will only be available if the service mesh sends the consumer NF FQDN in XFCC header, otherwise this value will be UNKNOWN in the metrics
ServiceOperation	Service operations as defined in 3gpp specification for NRF
Scope	Scope as received in the AccessToken Request
ResponseReason	Response Reason in Response sent back to NF
SubscriptionId	Subscription Id generated by OCNRF for NFStatusSubscribe Service Operation
NFType	Used in Gateway metrics. NF Type extracted from URI. Path is /n**xxx-yyy**/v**z**/..... Where **xxx will be changed to (Upper Case)** is NFType UNKNOWN if unable to extract NFType from the path Example: nnrf-nfm/v1/nf-instances

Table 10-1 (Cont.) Dimensions Legend

Dimension	Details
NFServiceType	Used in Gateway metrics. NF Type extracted from URI. Path is /**nxxx-yyy**/vz/..... Where **nxxx-yyy** is NFServiceType UNKNOWN if unable to extract NFServiceType from the path Example: nnrf-nfm/v1/nf-instances
Host	Used in Gateway metrics. (Ip or fqdn): port of gateway
HttpVersion	Http protocol version - HTTP/1.1, HTTP/2.0
Scheme	Http protocol scheme - HTTP, HTTPS, UNKNOWN
ClientCertIdentity	Used in Gateway metrics. Certificate Identity of the client, SAN=127.0.0.1,localhost CN=localhost, N/A if data is not available
Route_Path	Used in Gateway metrics. Path predicate/Header predicate that matched the current request
InstanceIdentifier	Used in Gateway metrics. Prefix of the pod configured in helm when there are multiple instances in same deployment- Prefix configured in helm otherwise UNKNOWN
ErrorOriginator	Used in Gateway metrics. This tag captures the ErrorOriginator - ServiceProducer, Nrf, IngressGW, None
Direction	Used in Gateway metrics. Direction of the request or response. egress, egressOut
error_reason	Reason for failure response received. If message is sent in the response, then it is filled with the message otherwise exception class is filled. In case of successful response it is filled with "no-error". Examples: error_reason="no_error" (In case successful response is received), error_reason="java.nio.channels.ClosedChannelException", error_reason="unable to find valid certification path to requested target"
KeyId	Key Id from Access Token Configuration used to sign the Access Token
KeyType	Key type of Access Token Configuration (private key or certificate)
isCurrentKeyId	True or False, when specific metric is for current key id in Access Token Configuration.

OCNRF Gateways Metrics

Table 10-2 OCNRF Gateways Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Dimensions
Total number of ingress requests	Total number of requests received at OCNRF	oc_ingressgateway_http_requests_total	Counter	Method NFType NFServiceType Host HttpVersion Scheme Route_path InstanceIdentifier ClientCertIdentity
NF Register Success	Total number of successful NFRegister service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="201 CREATED",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="PUT"}	Counter	Method NFType NFServiceType Host HttpVersion Scheme Route_path InstanceIdentifier ClientCertIdentity
NF Update Success (Complete Replacement)	Total number of successful NFUpdate service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="200 OK",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="PUT"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
NF Update Success (Partial Replacement)	Total number of successful NFUpdate service operations at OCNRF	oc_ingressgateway_http_responses_total{Status=~".*2.*",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="PATCH"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity

Table 10-2 (Cont.) OCNRF Gateways Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Dimensions
NF List/Profile Retrieval Success	Total number of successful NF List/ Profile retrieval service operations at OCNRF	oc_ingressgateway_http_responses_total{Status=~".*2.*",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="GET"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
Access Token Success	Total number of successful Access Token service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="200 OK",Route_path=~".*/oauth2/token*."}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
NF De-register Success	Total number of successful service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="204 NO_CONTENT",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="DELETE"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
NF Subscribe Success	Total number of successful NFStatusSubscribe service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="201 CREATED",Route_path=~".*nnrf-nfm/v1/subscriptions.*",Method="POST"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity

Table 10-2 (Cont.) OCNRF Gateways Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Dimensions
NF Unsubscribe Success	Total number of successful NFStatusUnSubscribe service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="204 NO_CONTENT",Route_path=~".*nnrf-nfm/v1/subscriptions.*",Method="DELETE"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
NF Discover Success	Total number of successful NFDiscover service operations at OCNRF	oc_ingressgateway_http_responses_total{Status=~"2.*",Route_path=~".*nnrf-disc/v1/nf-instances.*",Method="GET"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
4xx Responses (NF-Instances)	Total number of 4xx responses (NfRegister/ NfUpdate/NfDelete/ NfProfileRetrieval/ NfListRetrieval)	oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-nfm/v1/nf-instances.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
4xx Responses (Subscriptions)	Total number of 4xx responses (NFStatusSubscribe/ NFStatusUnSubscribe)	oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-nfm/v1/subscriptions.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity

Table 10-2 (Cont.) OCNRF Gateways Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Dimensions
4xx Responses (Discovery)	Total number of 4xx responses (NfDiscover)	oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-disc/v1/nf-instances.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
4xx Responses (AccessToken)	Total number of 4xx responses (NfAccessToken)	oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*oauth2/token.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
5xx Responses (NF-Instances)	Total number of 5xx responses (NfRegister/NfUpdate/NfDelete/NfProfileRetrieval/NfListRetrieval)	oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-nfm/v1/nf-instances.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
5xx Responses (Subscriptions)	Total number of 5xx responses (NFStatusSubscribe/NFStatusUnSubscribe)	oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-nfm/v1/subscriptions.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity

Table 10-2 (Cont.) OCNRF Gateways Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Dimensions
5xx Responses (Discovery)	Total number of 5xx responses (NfDiscover)	oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-disc/v1/nf-instances.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme Identifier ClientCertIdentity
5xx Responses (AccessToken)	Total number of 5xx responses (NfAccessToken)	oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*oauth2/token.*"}	Counter	Status Method Route_path NFType NFServiceType Host HttpVersion Scheme InstanceIdentifier ClientCertIdentity
Avg NRF Latency	Time (in microseconds) to process an ingress request. Measured from when the request is received to when the response is sent	oc_ingressgateway_request_latency_seconds	Timer	quantile InstanceIdentifier
Connection Failures Ingress Gateway	Metric to capture the connection failures when connect to the destination service fails. Here in case of Ingress gateway, the destination service will be a backend microservice of the NF. TLS connection failure metrics when connecting to ingress.	oc_ingressgateway_connection_failure_total	Counter	Host Port InstanceIdentifier error_reason
Ingress Gateway Request Processing Latency	Metric to capture the amount of time taken for processing of the request only within Ingress gateway.	oc_ingressgateway_request_processing_latency_seconds	Timer	quantile InstanceIdentifier

Table 10-2 (Cont.) OCNRF Gateways Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Dimensions
Total number of Egress requests	Metric to capture the request count reaches the Egress gateway from the application microservice and pegs with Direction as egress. Also, when the request goes out of egress gateway and pegs with Direction as egressOut.	oc_egressgateway_http_requests_total	Counter	Method NFType NFServiceType Host HttpVersion Scheme Proxy InstanceIdentifier Direction
Total number of Egress responses	Metric to capture when Egress gateway sends response back to backend NF microservice and pegs with Direction as egress. Also, when the response is received Egress gateway and pegs with Direction as egressOut.	oc_egressgateway_http_responses_total	Counter	Status Method NFType NFServiceType Host HttpVersion Scheme InstanceIdentifier Direction
Connection Failures Egress Gateway	Metric to capture failure while connecting the backend microservice and destination service	oc_egressgateway_connection_failure_total	Counter	Host Port InstanceIdentifier Direction error_reason
Egress Gateway Request Processing Latency	Metric captures the amount of time taken for processing of the request only within Egress gateway.	oc_egressgateway_request_processing_latency_seconds	Timer	quantile InstanceIdentifier

OCNRF NF Metrics

Table 10-3 OCNRF NF Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Recommended legend to see dimension level data (as applicable)	Dimensions
NfRegistrations Total	Number of Registration Requests received	ocnrf_nfRegister_rx_requests_total	Counter	NfRegistrations Total	NrfLevel, NfInstanceId, RequesterNfType, NfFqdn

Table 10-3 (Cont.) OCNRF NF Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Recommended legend to see dimension level data (as applicable)	Dimensions
NfRegistrations Responses Total	Number of Registration Responses sent.	ocnrf_nfRegister_tx_responses_total	Counter	NfRegistrations Responses Total	NrfLevel, NfInstanceId, RequesterNfType, HttpStatusCode, NfFqdn
NfRegistrations Per Service Total	Number of Registrations received and processed successfully per Service.	ocnrf_nfRegister_rx_requests_success_perService_total	Counter	NfRegistrations Per Service [serviceName :- {{ serviceName }}, nfInstanceId :- {{NfInstanceId}}]	NrfLevel, NfInstanceId, ServiceName, ServiceInstanceId, NfFqdn
NFUpdates Total	Number of Update Requests received.	ocnrf_nfUpdate_rx_requests_total	Counter	NfUpdates Total	NrfLevel, NfInstanceId RequesterNfType, UpdateType (Partial/Complete), HttpStatusCode, NfFqdn
NFUpdates Responses Total	Number of Update Responses sent.	ocnrf_nfUpdate_tx_responses_total	Counter	NfUpdates Responses Total	NrfLevel, NfInstanceId, RequesterNfType, UpdateType (Partial/Complete), HttpStatusCode, NfFqdn
NFUpdates Per Service Total	Number of NfUpdates received and processed successfully per Service.	ocnrf_nfUpdate_rx_requests_success_perService_total	Counter	NFUpdates Per Service [serviceName :- {{ serviceName }}, serviceInstanceId:- {{ServiceInstanceId}}]	NrfLevel, Updatetype =(Partial/Complete), NfInstanceId, ServiceName, ServiceInstanceId, NfFqdn
Heartbeat Requests Total	Number of Heartbeat Requests received	ocnrf_nfHeartbeat_rx_requests_total	Counter		NrfLevel, NfInstanceId, RequesterNfType, NfFqdn
Heartbeat Responses Total	Number of Heartbeat Responses sent	ocnrf_nfHeartbeat_tx_responses_total	Counter		Nrflevel, NfInstanceId, RequesterNfType, HttpStatusCode, NfFqdn
NF De-Registration Requests Total	Number of De-registration requests received	ocnrf_nfDeregister_rx_requests_total	Counter		NrfLevel, NfInstanceId, RequesterNfType, NfFqdn
NF De-Registration Responses Total	Number of De-registration responses sent	ocnrf_nfDeregister_tx_responses_total	Counter		NrfLevel, NfInstanceId, RequesterNfType, HttpStatusCode, NfFqdn

Table 10-3 (Cont.) OCNRF NF Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Recommended legend to see dimension level data (as applicable)	Dimensions
NF De-Registrations Per Service Total	Number of De-registration requests received and process successfully per Service	ocnrf_nfDeregister_rx_requests_success_per Service_total	Counter	NFDeregistration Per Service [serviceName :- {{ serviceName }}, serviceInstanceId:- {{ServiceInstanceId}}]	NrfLevel, ServiceName, ServiceInstanceId, NfInstanceId, NfFqdn
NF List Retrieval Requests Total	Number of NFListRetrieval requests received	ocnrf_nfListRetrieval_rx_requests_total	Counter		NrfLevel, RequesterNfType, NfFqdn
NF List Retrieval Responses Total	Number of NFListRetrieval responses sent	ocnrf_nfListRetrieval_tx_responses_total	Counter		NrfLevel, RequesterNfType, HttpStatusCode, NfFqdn
NF Profile Retrieval Requests Total	Number of NFProfileRetrieval requests received	ocnrf_nfProfile Retrieval_rx_requests_total	Counter		NrfLevel, NfInstanceId, NfFqdn
NF Profile Retrieval Responses Total	Number of NFProfileRetrieval responses sent	ocnrf_nfProfile Retrieval_tx_responses_total	Counter		NrfLevel, NfInstanceId, HttpStatusCode, NfFqdn
Number of Heartbeats missed	Number of heartbeats missed.	ocnrf_heartbeat_missed_total	Counter		NrfLevel, NfType, NfInstanceId, NfFqdn
NF Status Subscribe Requests Total	Number of NStatusSubscribe requests received	ocnrf_nfStatus Subscribe_rx_requests_total	Counter		NrfLevel, RequesterNfType, OperationType, NfFqdn
NF Status Subscribe Responses Total	Number of NfStatusSubscribe responses sent	ocnrf_nfStatus Subscribe_tx_responses_total	Counter		NrfLevel, RequesterNfType, HttpStatusCode, OperationType, NfFqdn
NF Status UnSubscribe Requests Total	Number of NfStatusUnsubscribe requests received	ocnrf_nfStatus Unsubscribe_rx_requests_total	Counter		NrfLevel, RequesterNfType, NfFqdn
NF Status UnSubscribe Responses Total	Number of NfStatusUnsubscribe responses sent	ocnrf_nfStatus Unsubscribe_tx_responses_total	Counter		NrfLevel, RequesterNfType, HttpStatusCode, NfFqdn
NF Status Notifications Requests Sent	Number of NfStatusNotify requests sent	ocnrf_nfStatus Notify_tx_requests_total	Counter		NrfLevel, NotificationEventType, TargetNfType, NfFqdn, SubscriptionId

Table 10-3 (Cont.) OCNRF NF Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Recommended legend to see dimension level data (as applicable)	Dimensions
NF Status Notifications Responses Received	Number of NfStatusNotify responses received	ocnrf_nfStatusNotify_rx_responses_total	Counter		NrfLevel, NotificationEventType, TargetNfType, HttpStatusCode, NfFqdn, SubscriptionId
NF Status Notifications Requests Failed	Number of NfStatusNotify requests failed to sent out	ocnrf_nfStatusNotify_requests_failed_total	Counter		NrfLevel, NotificationEventType, TargetNfType, NfFqdn,SubscriptionId
NfDiscover Requests Total	Number of NfDiscover Requests received	ocnrf_nfDiscover_rx_requests_total	Counter	NfDiscover Req [TargetNf :- {{ TargetNfType }}, RequesterNfType :- {{RequesterNfType}}]	NrfLevel, TargetNfType, RequesterNfType, NfFqdn
NfDiscover Responses Total	Number of NfDiscover responses sent	ocnrf_nfDiscover_tx_responses_total	Counter		NrfLevel, TargetNfType, RequesterNfType, HttpResponseCode, NfFqdn
NFDiscover Per Service Total	Number of NfDiscover requests received and processed successfully per Service	ocnrf_nfDiscover_rx_requests_success_perService_total	Counter	NFDiscover Per Service [serviceName :- {{ serviceName }}]	NrfLevel, RequesterNfType, ServiceName, NfFqdn
Discovered profiles	Number of Profiles returned in discovery response. Depending on bucket size and corresponding value will tell how many profiles are returned in discovery response.	ocnrf_nfDiscover_profiles_discovered_total	Counter	Discovered profiles [TargetNfType :- {{TargetNfType}}, Bucket :- {{ Bucket }}]	NrfLevel, TargetNfType, BucketSize, NfFqdn
Active Registrations	Number of active registered NFs at any point of time	ocnrf_active_registrations_count	Gauge	Active Registrations [NfType-{{ NfType }}, NrfLevel- {{ NrfLevel }}]	NfType, NrfLevel

Table 10-3 (Cont.) OCNRF NF Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Recommended legend to see dimension level data (as applicable)	Dimensions
Avg NRF Latency taken by NRF specific microservice	Time taken by NRF specific microservice to process the service operation (NfRegister/ NfUpdate/ NfDelete/ NfProfileRetrieval/ NfListRetrieval/ NfHeartbeat/ NfDiscover/ NFStatusSubscribe/ NFStatusUnSubscribe/ NfAccessToken) **Note**: Latency calculated by this metric doesn't include time taken by OCNRF API gateway.	ocnrf_message_processing_time_seconds	Timer	Avg NRF Latency {{ ServiceOperation } } {{ RequesterNfType } }	NrfLevel, RequesterNfType, ServiceOperation
OCNRF database operations	Database operation count corresponding to every service operation	ocnrf_dbmetric_total	Counter		Method, DBOperation, NrfLevel, HttpStatusCode
Database operation round trip time	Time (in microseconds) taken by database operation corresponding to every service operation NfRegister/ NfUpdate/ NfDelete/ NfProfileRetrieval/ NfListRetrieval/ NfHeartbeat/ NfDiscover/ NFStatusSubscribe/ NFStatusUnSubscribe/ NfAccessToken)	ocnrf_dbmetrics_round_trip_time_seconds	Timer		Method, DBOperation, ServiceOperation, TableName: (NRF Table Names), NrfLevel, HttpStatusCode

NF Screening Metrics

Table 10-4 NF Screening metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
Total NF Requests for which Screening Failed	The total number of requests for which screening failed against NF FQDN screening list.	ocnrf_nfScreening_nfFqdn_requestFailed_total	Counter	NFRegister, NFUpdate	NRF level, NF type, NfFqdn
Total NF Requests Rejected due to Screening Failed	The total number of requests rejected because screening failed against NF FQDN screening list.	ocnrf_nfScreening_nfFqdn_requestRejected_total	Counter	NFRegister, NFUpdate	NRF level, NF type, NfFqdn
Total NF Requests for which Screening Failed	The total number of requests for which screening failed against NF IP endpoint screening list.	ocnrf_nfScreening_nfIpEndPoint_requestFailed_total	Counter	NFRegister, NFUpdate	NRF level, NF type, NfFqdn
Total NF Requests Rejected due to Screening Failed	The total number of requests rejected because screening failed against NF IP endpoint screening list.	ocnrf_nfScreening_nfIpEndPoint_requestRejected_total	Counter	NFRegister, NFUpdate	NRF level NF type NfFqdn
Total NF Requests for which Screening Failed	The total number of requests for which screening failed against Callback URI screening list.	ocnrf_nfScreening_callbackUri_requestFailed_total	Counter	NFRegister, NFUpdate, NFStatusSubscribe	NRF level, NF type, NfFqdn
Total NF Requests Rejected due to Screening Failed	The total number of requests rejected because screening failed against Callback URI screening list.	ocnrf_nfScreening_callbackUri_requestRejected_total	Counter	NFRegister, NFUpdate, NFStatusSubscribe	NRF level, NF type, NfFqdn
Total NF Requests for which Screening Failed	The total number of requests for which screening failed against PLMN id screening list.	ocnrf_nfScreening_plmnId_requestFailed_total	Counter	NFRegister, NFUpdate	NRF level NF type NfFqdn
Total NF Requests Rejected due to Screening Failed	The total number of requests rejected because screening failed against PLMN id screening list.	ocnrf_nfScreening_plmnId_requestRejected_total	Counter	NFRegister, NFUpdate	NRF level, NF type, NfFqdn
Total NF Requests for which Screening Failed	The total number of NFRegister requests rejected as NF type was not allowed to register with NRF.	ocnrf_nfScreening_nfTypeRegister_requestFailed_total	Counter	NFRegister	NRF level, NF type, NfFqdn

Table 10-4 (Cont.) NF Screening metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
Total NF Requests Rejected due to Screening Failed	The total number of NFRegister requests for which screening failed against NF type screening list.	ocnrf_nfScreening_nfTypeRegister_requestRejected_total	Counter	NFRegister	NRF level, NF type, NfFqdn
NF Screening not applied Internal Error	The total number of times screening not applied due to internal error.	ocnrf_nfScreening_notApplied_InternalError_total	Counter	NFRegister, NFUpdate, NFStatusSubscribe	NRF level, NF type, NfFqdn

Note:

In the above "NF Screening metrics" table, the dimension NF Type is a requester NF Type.

NF Access token Metrics

Table 10-5 NF Access token metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Access Token Request Received Total	The total number of access token requests received	ocnrf_accessToken_rx_requests_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, NfFqdn
NF Access Token Responses Sent Total	The total number of access token responses sent	ocnrf_accessToken_tx_responses_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType
NF Access Token Request Rejected (ClientNotAuthorized)	Number of access token request for which client authorized failed.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, RejectionReason HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ClientNotAuthorized

Table 10-5 (Cont.) NF Access token metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Access Token Request Rejected (ProducerWithRequestedScopeNotFound)	Number of access token not granted because of no producer instance registered for service/s in the scope.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, RejectionReason HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ProducerWithRequestedScopeNotFound
NF Access Token Request Rejected (ProducerWithRequestedNfInstanceIdNotFound)	Number of access token not granted because of no producer instance registered for No producer instance is registered at all for provided target Instance Id in request.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, RejectionReason HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ProducerWithRequestedNfInstanceIdNotFound
NF Access Token Request Rejected (InconsistentScope)	Number of access token not granted because services in the scope belong to different NF types.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, RejectionReason HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = InconsistentScope
NF Access Token Request Rejected (ConsumerNFTypeMismatch)	Number of access token not granted because consumer NF type in profile is not matching with the access token request.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, RejectionReason HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ConsumerNFTypeMismatch

Table 10-5 (Cont.) NF Access token metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Access Token Request Rejected (ProducerNFTypeMismatch)	Number of access token not granted because producer NF type in profile is not matching with the access token request.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, RejectionReason HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ProducerNFTypeMismatch
NF Access Token Request Rejected (InternalError)	Number of access token not granted because failure at NRF due to internal error.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, RejectionReason HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ProducerNFTypeMismatch
NF Access Token Request Rejected (ConsumerNfTypeNotAllowed)	Number of access token not granted because the consumer NFType is not allowed to access the requested NF.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ConsumerNfTypeNotAllowed
NF Access Token Request Rejected (ConsumerPlmnNotAllowed)	Number of access token not granted because the consumer NF PLMN is not allowed to access the requested NF.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ConsumerPlmnNotAllowed
NF Access Token Request Rejected (SecretNotAccessible)	Number of access token not granted because the secret for current key id is not accessible.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = SecretNotAccessible

Table 10-5 (Cont.) NF Access token metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Access Token Request Rejected (InvalidFileData)	Number of access token not granted because the current key id file data is invalid.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = InvalidFileData
NF Access Token Request Rejected (NamespaceNotAccessible)	Number of access token not granted because the namspace for current key id is not accessible.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = NamespaceNotAccessible
NF Access Token Request Rejected (FileNotFound)	Number of access token not granted because the file not found in secrets.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = FileNotFound
NF Access Token Request Rejected (CurrentKeyIdNotConfigured)	Number of access token not granted because the current key id is not configured.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = CurrentKeyIdNotConfigured
NF Access Token Request Rejected (ExpiredCertificate)	Number of access token not granted because the OCNRF certificate is expired.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = ExpiredCertificate
NF Access Token Request Rejected (BadRequest)	Number of access token not granted because the Request is incorrect.	ocnrf_accessToken_tx_rejected_total	Counter	AccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode, NfFqdn, KeyId, KeyType RejectionReason = BadRequest

NRF Configuration Metrics

Table 10-6 NRF Configuration Metrics

Metric Name	Metric Details	Metric Filter	Metric Type	Service Operation	Dimensions
OCNRF Oauth Token Signing Keys Health Status	Oauth Token Signing keys health status	ocnrf_oauth_keyData_healthStatus Value 0 - Healthy Value 1 - Unhealthy	Gauge	Configuration	KeyId, KeyType, isCurrentKeyId, NrfLevel
OCNRF Oauth Current KeyId Configuration Status	Oauth Current Key Id Configuration Status	ocnrf_oauth_currentKeyId_configuredStatus Value 0 - Healthy Value 1 - Unhealthy	Gauge	Configuration	NrfLevel
OCNRF Oauth Token Signing Keys Expiry Status	Oauth Token Signing keys Expiry Status	ocnrf_oauth_keyData_expiryStatus (Value is expiry time in epoch time)	Gauge	Configuration	KeyId, isCurrentKeyId, NrfLevel

NRF-SLF Metrics

Table 10-7 NRF-SLF metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
Discover Request Received For SLF Total	The total number of NF Discover request received for SLF	ocnrf_nfDiscover_ForSLF_rx_requests_total	Counter	NFDiscover	TargetNfType, NRFLevel, NfFqdn

Table 10-7 (Cont.) NRF-SLF metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
Discover Response Sent For SLF Total	The total number of NF Discover responses sent for SLF	ocnrf_nfDiscover_ForSLF_tx_responses_total	Counter	NFDiscover	TargetNfType, NRFLevel, HttpStatusCode, ResponseReason, NfFqdn Possible Response Reasons: ResponseReason = SLFCommunicationFailure ResponseReason = MandatoryParamsMissing ResponseReason = SLFSubscriberNotProvisioned ResponseReason = ErrorFromSLF ResponseReason = InternalError ResponseReason = SuccessFromSLF ResponseReason = GroupIdUsedFromSearchQuery
SLF Query Requests Sent Total	The total number of SLF query request sent	ocnrf_SLF_tx_requests_total	Counter	NFDiscover	TargetNfType, NRFLevel, SubscriptionIdType, NfFqdn
SLF Query Responses Received Total	The total number of SLF query response received	ocnrf_SLF_rx_responses_total	Counter	NFDiscover	TargetNfType, NRFLevel, SubscriptionIdType,HttpStatusCode, GroupId, NfFqdn

Table 10-7 (Cont.) NRF-SLF metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
SLF Round Trip Time Total	Time (in microseconds) after sending query to SLF and getting response from SLF	ocnrf_slf_round_trip_time_seconds	Timer	NFDiscover	TargetNfType, SubscriptionIdType, HttpStatusCode, GroupId, NrfLevel, SLF ApiRoot, NfFqdn

NRF Forwarding Metrics

Table 10-8 NRF Forwarding Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Access Token Requests Forwarded Total	The total number of Access Token Request forwarded to Primary/Secondary NRF	ocnrf_forward_accessToken_tx_requests_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel, NfFqdn

Table 10-8 (Cont.) NRF Forwarding Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Access Token Forwarded Responses Total	The total number of Access Token Responses for request forwarded to Primary/Secondary NRF	ocnrf_forward_accessToken_rx_responses_total	Counter	AccessToken	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceName, NrfLevel,HttpStatusCode, RejectionReason, NfFqdn **RejectionReason**: • InternalError • NRFCommunicationFailure • ErrorFromNRF • NRFForwardingConfigurationMissing • LoopDetected *NotApplicable is applicable for 2xx Status code
NF Profile Retrieval Requests Forwarded Total	The total number of Profile Retrieval Request forwarded to Primary/Secondary NRF	ocnrf_forward_nfProfileRetrieval_tx_requests_total	Counter	NFProfile Retrieval	NrfLevel, NfInstanceId, NfFqdn

Table 10-8 (Cont.) NRF Forwarding Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Profile Retrieval Forwarded Responses Total	The total number of Profile Retrieval Responses for Request forwarded to Primary/Secondary NRF	ocnrf_forward_nfProfileRetrieval_rx_responses_total	Counter	NFProfile Retrieval	NrfLevel, NfInstanceId, HttpStatusCode, RejectionReason, NfFqdn **RejectionReason**: • InternalError • NRFCommunicationFailure • ErrorFromNRF • NRFForwardingConfigurationMissing • LoopDetected *NotApplicable is applicable for 2xx Status code
NF Status Subscribe Forwarded Requests Total	The total number of Status Subscribe Request forwarded to Primary/Secondary NRF	ocnrf_forward_nfStatusSubscribe_tx_requests_total	Counter	NFStatus Subscribe, NFStatus Unsubscribe	NrfLevel, RequesterNfType, OperationType, NfFqdn

Table 10-8 (Cont.) NRF Forwarding Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Status Subscribe Forwarded Responses Total	The total number of Responses for Status Subscribe Request forwarded to Primary/ Secondary NRF	ocnrf_forward_nfStatusSubscribe_rx_responses_total	Counter	NFStatus Subscribe, NFStatus Unsubscribe,	NrfLevel, RequesterNfType, HttpStatusCode, OperationType, RejectionReason, NfFqdn **RejectionReason**: • InternalError • NRFCommunicationFailure • ErrorFromNRF • NRFForwardingConfigurationMissing • LoopDetected *NotApplicable is applicable for 2xx Status code
NF Discovery Forwarded Requests Total	The total number of NF Discovery Request forwarded to Primary/Secondary NRF	ocnrf_forward_nfDiscover_tx_requests_total	Counter	NFDiscover	NrfLevel, TargetNfType, RequesterNfType, NfFqdn

Table 10-8 (Cont.) NRF Forwarding Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Discovery Forwarded Responses Total	The total number of Responses for NF Discovery Request forwarded to Primary/ Secondary NRF	ocnrf_forward_nfDiscover_rx_responses_total	Counter	NFDiscover	NrfLevel, TargetNfType, RequesterNfType, HttpResponseCode, RejectionReason, NfFqdn **RejectionReason**: • InternalError • NrfCommunicationFailure • NrfForwardingConfigurationMissing • LoopDetected ErrorFromNrf *NotApplicable is applicable for 2xx Status code
Avg Latency for NRF Message Forwarding	Time taken by NRF specific microservice to forward the message to other Primary/Secondary NRF with the service operation: (NFProfileRetrieval/ NFDiscover/ NFStatusSubscribe/ NfStatusUnsubscribe /AccessToken)	ocnrf_forward_round_trip_time_seconds	Timer	NFStatusSubscribe, NFStatusUnsubscribe, NFProfileRetrieval, NFDiscover, AccessToken	NrfLevel, RequesterNfType, ServiceOperation, NfFqdn

GeoRedundancy metrics

Table 10-9 GeoRedundancy metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
DB Replication status	The current replication status of the DBTier service. This metric is pegged only if the GeoRedundancy Feature is enabled.	ocnrf_dbreplication_status	Gauge	NA	NrfLevel, DbReplicationStatus
DB Replication down Time	Time taken for the replication status to change from "INACTIVE" to "ACTIVE". This metric is pegged only if the GeoRedundancy Feature is enabled.	ocnrf_dbreplication_down_time_seconds	Timer	NA	NrfLevel,DbReplicationDownStartTime,DbReplicationDownEndTime
Total NfInstances switched over from mated site	The number of NFInstances that got switched over from the mated site.	ocnrf_nf_switch_over_total	Counter	NfRegister, NfUpdate,NfDeregister, NfHeartbeat	NrfLevel, NfInstanceId,RemoteNrfInstanceId,ServiceOperation,OperationType, NfFqdn
Total NfSubscriptions switched over from mated site	The number of NfSubscriptions that got switched over from the mated site.	ocnrf_nfSubscriptions_switch_over_total	Counter	NfStatusSubscribe,NfStatusUnsubscribe, NrfAuditor	NrfLevel,SubscriptionId,RemoteNrfInstanceId, ServiceOperation,OperationType
Total Nfinstances removed by OCNRF as it is stale	The number of NfInstances that get deleted by the NrfAuditor when it detects a record to be stale.	ocnrf_stale_nf_deleted_total	Counter	NA	NrfLevel, NfInstanceId, NfStatus, NfFqdn
Total NfSubscriptions removed by OCNRF as it is stale	The number of NfSubscriptions that get deleted by the NrfAuditor when it detects a record to be stale.	ocnrf_stale_nfSubscriptions_deleted_total	Counter	NA	NrfLevel,NfSubscriptionId,SubscriptionStatus
Total NfInstances that have been marked as SUSPENDED by the OCNRF Auditor	The number of profiles that have been marked as SUSPENDED when a profile has missed nfHeartBeatMissAllowed.	ocnrf_nf_suspended_total	Counter	NA	NrfLevel, NfInstanceId,NfStatus, HeartbeatTimer, NfFqdn

Table 10-9 (Cont.) GeoRedundancy metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
Total NfSubscriptions whose validityTime has expired	The number of NfSubscriptions whose validityTime has expired	ocnrf_nfSubscriptions_expired_total	Counter		NrfLevel, SubscriptionId

NF AccessToken Authorization Metrics

Table 10-10 NF AccessToken Authorization Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Access Token Request Rejected (AuthScreeningFailed)	Number of access token not granted because the consumer NF is not authorized to access the requested NF or its services.	ocnrf_accessToken_tx_rejected_total	Counter	NfAccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, ServiceNameScope, NrfLevel, NfFqdn, HttpStatusCode RejectionReason = ClientNotAuthorized

NF Authentication Metrics

Table 10-11 NF Authentication Metrics

Metric Name	Metric Details	Metric filter	Metric Type	Service Operation	Dimensions
NF Authentication Failure Total	The total number of request for which FQDN based Authentication failed at OCNRF	ocnrf_nf_authentication_failure_total	Counter	NrfLevel, Method, ServiceOperation, NfFqdn, TLSFqdn	NFAccessToken/ NFRegistration/ NFSubscription /NFDiscovery/ NfListRetrieval/ NfProfileRetrieval For NfListRetrieval and NfProfileRetrieval serviceOperations NfFqdn is filled as NotApplicable. If OC-XFCC-DNS header is not received at NRF Microservice then TLSFqdn is filled as "UNKNOWN".

OCNRF KPIs

This section includes information about KPIs for Oracle Communications Network Repository Function (OCNRF).

> **Note:**
>
> Sample OCNRF dashboard for Grafana is delivered to the customer through OCNRF Custom Templates. Metrics and functions used to achieve KPI are already covered in OCNRF Custom Templates.

Table 10-12 KPI Details

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response code
OCNRF Ingress Request	Rate of HTTP requests received at OCNRF Ingress Gateway	oc_ingressgateway_http_requests_total	All	Not Applicable
NF Register Success		sum(irate(oc_ingressgateway_http_responses_total{Status="201 CREATED",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="PUT"}[5m])) sum(irate(oc_ingressgateway_http_responses_total{Status="201 CREATED",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="PUT"}[5m]))	NFRegister	201
NF Update Success (Complete Replacement)		sum(irate(oc_ingressgateway_http_responses_total{Status="200 OK",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="PUT"}[5m]))	NFUpdate	200
NF DeRegister Success		sum(irate(oc_ingressgateway_http_responses_total{Status="204 NO_CONTENT",Route_path=~".*nnrf-nfm/v1/nf-instances.*",Method="DELETE"}[5m]))	NFDeregister	204
NF Subscribe Success		sum(irate(oc_ingressgateway_http_responses_total{Status="201 CREATED",Route_path=~".*nnrf-nfm/v1/subscriptions.*",Method="POST"}[5m]))	NFStatusSubscribe	201
NF Unsubscribe Success		sum(irate(oc_ingressgateway_http_responses_total{Status="204 NO_CONTENT",Route_path=~".*nnrf-nfm/v1/subscriptions.*",Method="DELETE"}[5m]))	NFStatusUnsubscribe	204
NF Discover Success		sum(irate(oc_ingressgateway_http_responses_total{Status=~"2.*",Route_path=~".*nnrf-disc/v1/nf-instances.*",Method="GET"}[5m]))	NFDiscover	200
4xx Responses (NF-Instances)		sum(irate(oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-nfm/v1/nf-instances.*"}[5m]))	NFRegister/ NFUpdate/ NFDeregister	4xx

Table 10-12 (Cont.) KPI Details

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response code
4xx Responses (Subscriptions)		sum(irate(oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-nfm/v1/subscriptions.*"}[5m]))	NFStatusSubscribe/ NFStatusUnsubscribe	4xx
4xx Responses (Discovery)		sum(irate(oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-disc/v1/nf-instances.*"}[5m]))	NFDiscover	4xx
5xx Responses (NF-Instances)		sum(irate(oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-nfm/v1/nf-instances.*"}[5m]))	NFRegister/ NFUpdate/ NFDeregister	5xx
5xx Responses (Subscriptions)		sum(irate(oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-nfm/v1/subscriptions.*"}[5m]))	NFStatusSubscribe/ NFStatusUnsubscribe	5xx
5xx Responses (Discovery)		sum(irate(oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-disc/v1/nf-instances.*"}[5m]))	NFDiscover	5xx

OCNRF Alerts

This section includes information about alerts for OCNRF.

Table 10-13 Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
System Level Alerts							

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfNfStatusUnavailable	All the OCNRF services are unavailable, either because the OCNRF is getting deployed or purged. These OCNRF services considered are nfregistration, nfsubscription, nrfauditor, nrfconfiguration, nfaccesstoken, nfdiscovery, appinfo, ingressgateway and egressgateway	Critical	**description**: 'OCNRF services unavailable' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : All OCNRF services are unavailable.'	1.3.6.1.4.1.323.5.3.36.1.2.7016	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared automatically when the OCNRF services start becoming available. Steps: 1. Check for service specific alerts. 2. Refer the application logs on Kibana and check for database related failures like connectivity, invalid secrets etc. The logs can be filtered based on the services. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfPodsRestart	A pod belonging to any of the OCNRF services have restarted.	Major	**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'	1.3.6.1.4.1.323.5.3.36.1.2.7017	'kube_pod_container_status_restarts_total' **Note**: This is a kubernetes metric. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared automatically if the specific pod is up. Steps: 1. Refer the application logs on Kibana and filter based on pod name, check for database related failures like connectivity, kubernetes secrets etc. 2. Check orchestration logs for liveness or readiness probe failures. 3. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
NnrfNFManagementServiceDown	Either NFRegistration or NFSubscription or NrfAuditor services are unavailable.	Critical	**description**: 'OCNRF Nnrf_Management service *<nfregistration|nfsubscription|nrfauditor>* is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFManagement service is down'	1.3.6.1.4.1.323.5.3.36.1.2.7018	"up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when all the Nnrf_NFManagement services are available that is nfregistration, nfsubscription and nrfauditor. Steps: 1. Check if NfService specific alerts are generated to understand which service is down. 2. Check the orchestration logs of nfregistration, nfsubscription and nrfauditor services and check for liveness or readiness probe failures. 3. Refer the application logs on Kibana and filter based on above service names. Check for ERROR WARNING logs for each of these services. 4. Refer the application logs on Kibana and filter the service appinfo, check for the service status of the nfregistration, nfsubscription and nrfauditor services. 5. Depending on the failure reason, take the resolution steps. 6. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
NnrfAccessTokenServiceDown	NFAccessToken service is unavailable.	Critical	**description**: 'OCNRF Nnrf_NFAccessToken service nfaccesstoken is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFAccessToken service down'	1.3.6.1.4.1.323.5.3.36.1.2.7020	"up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available use the similar metric as exposed by the monitoring system.	The alert is cleared when the Nnrf_AccessToken service is available. Steps: 1. Check the orchestration logs of nfaccesstoken service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nfaccesstoken service names. Check for ERROR WARNING logs. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
NnrfNFDiscoveryServiceDown	NFDiscovery is unavailable.	Critical	**description**: 'OCNRF Nnrf_NFDiscovery service *nfdiscovery* is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFDiscovery service down'	1.3.6.1.4.1.323.5.3.36.1.2.7019	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the Nnrf_NFDiscovery service is available. Steps: 1. Check the orchestration logs of nfdiscovery service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nfdiscovery service names. Check for ERROR WARNING logs. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfRegistrationServiceDown	None of the pods of the NFRegistration microservice is available.	Critical	**description**: 'OCNRF NFRegistration service *nfregistration* is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFRegistration service is down'	1.3.6.1.4.1.323.5.3.36.1.2.7021	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the nfregistration service is available. Steps: 1. Check the orchestration logs of nfregistration service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nfregistration service names. Check for ERROR WARNING logs. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfSubscriptionServiceDown	None of the pods of the NFSubscription microservice is available.	Critical	**description**: 'OCNRF NFSubscription service *nfsubscription* is down**.** **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFSubscription service is down'	1.3.6.1.4.1.323.5.3.36.1.2.7022	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the nfsubscription service is available. Steps: 1. Check the orchestration logs of nfsubscription service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nfsubcription service names. Check for ERROR WARNING logs. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfDiscoveryService Down	None of the pods of the NFDiscovery microservice is available.	Critical	**description**: 'OCNRF NFDiscovery service *nfdiscovery* is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFDiscovery service down'	1.3.6.1.4.1.323.5.3.36.1.2.7023	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the nfdiscovery service is available. Steps: 1. Check the orchestration logs of nfdiscovery service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nfdiscovery service names. Check for ERROR WARNING logs. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfAccessTokenServiceDown	None of the pods of the NFAccessToken microservice is available.	Critical	**description**: 'OCNRF NFAccessToken service *nfaccesstoken* is down **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFAccesstoken service down'	1.3.6.1.4.1.323.5.3.36.1.2.7024	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the nfaccesstoken service is available. Steps: 1. Check the orchestration logs of nfaccesstoken service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nfaccesstoken service names. Check for ERROR WARNING logs. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfAuditorServiceDown	None of the pods of the NrfAuditor microservice is available.	Critical	**description**: 'OCNRF NrfAuditor service *nrfauditor* is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NrfAuditor service down'	1.3.6.1.4.1.323.5.3.36.1.2.7026	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the nrfauditor service is available. Steps: 1. Check the orchestration logs of nrfauditor service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nrfauditor service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfConfigurationServiceDown	None of the pods of the NrfConfiguration microservice is available.	Critical	**description**: 'OCNRF NrfConfiguration service *nrfconfiguration* is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NrfConfiguration service down'	1.3.6.1.4.1.323.5.3.36.1.2.7025	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the nrfconfiguration service is available. Steps: 1. Check the orchestration logs of nrfconfiguration service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on nrfconfiguration service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfAppInfoServiceDown	None of the pods of the App Info microservice is available.	Critical	**description**: 'OCNRF Appinfo service *appinfo* is down' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : Appinfo service down'	1.3.6.1.4.1.323.5.3.36.1.2.7027	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the app-info service is available. Steps: 1. Check the orchestration logs of appinfo service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on appinfo service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfIngressGatewayServiceDown	None of the pods of the Ingress-Gateway microservice is available.	Critical	**description**: 'OCNRF Ingress-Gateway service *ingressgateway* is down**.** **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : Ingress-gateway service down'	1.3.6.1.4.1.323.5.3.36.1.2.7028	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the ingressgateway service is available. Steps: 1. Check the orchestration logs of ingress-gateway service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on ingress-gateway service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfEgressGatewayServiceDown	None of the pods of the Egress-Gateway microservice is available.	Critical	**description**: 'OCNRF Egress-Gateway service *egressgateway* 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'	1.3.6.1.4.1.323.5.3.36.1.2.7029	'up' **Note**: This is a prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.	The alert is cleared when the egressgateway service is available. **Note**: The threshold is configurable in the alerts.yaml Steps: 1. Check the orchestration logs of egress-gateway service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on egress-gateway service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfMemoryUsageCrossedMinorThreshold	A pod has reached the configured minor threshold(50%) of its memory resource limits.	Minor	**description**: 'OCNRF Memory Usage for pod <*Pod name*> 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.'	1.3.6.1.4.1.323.5.3.36.1.2.7030	'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.	The alert gets cleared when the memory utilization falls below the Minor Threshold or crosses the major threshold, in which case OcnrfMemoryUsageCrossedMajorThreshold alert shall be raised. **Note**: The threshold is configurable in the alerts.yaml If guidance required, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfMemoryUsageCrossedMajorThreshold	A pod has reached the configured major threshold(60%) of its memory resource limits.	Major	**description**: 'OCNRF Memory Usage for pod <*Pod name*> has crossed the 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.'	1.3.6.1.4.1.323.5.3.36.1.2.7031	'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.	The alert gets cleared when the memory utilization falls below the Major Threshold or crosses the critical threshold, in which case OcnrfMemoryUsageCrossedCriticalThreshold alert shall be raised. **Note**: The threshold is configurable in the alerts.yaml If guidance required, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfMemoryUsageCrossedCriticalThreshold	A pod has reached the configured critical threshold (70%) of its memory resource limits.	Critical	**description**: 'OCNRF Memory Usage for pod <*Pod name*> has crossed the configured critical threshold (70%) (value = {{ $value }}) of its limit.' **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : Memory Usage of pod exceeded 70% of its limit.'	1.3.6.1.4.1.323.5.3.36.1.2.7032	'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.	The alert gets cleared when the memory utilization falls below the Critical Threshold. **Note**: The threshold is configurable in the alerts.yaml If guidance required, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTotalIngressTrafficRateAboveMinorThreshold	The total OCNRF Ingress Message rate has crossed the configured minor threshold of 800 TPS. Default value of this alert trigger point in NrfAlertValues.yaml is when OCNRF Ingress Rate crosses 80 % of 1000 (Maximum ingress request rate)	Minor	**description**: Total'Ingress traffic Rate is above configured minor threshold i.e. 800 requests per second (current value is: {{ $value }})' **summary**: 'timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)'	1.3.6.1.4.1.323.5.3.36.1.2.7001	'oc_ingressgateway_http_requests_total'	The alert is cleared either when the total Ingress Traffic rate falls below the Minor threshold or when the total traffic rate cross the Major threshold, in which case the OcnrfTotalIngressTrafficRateAboveMinorThreshold alert shall be raised. **Note**: The threshold is configurable in the alerts.yaml Steps: Reassess why the OCNRF is receiving additional traffic (for example: geo redundancy OCNRF is unavailable). If this is unexpected, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTotalIngressTrafficRateAboveMajorThreshold	The total OCNRF Ingress Message rate has crossed the configured major threshold of 900 TPS. Default value of this alert trigger point in NrfAlertValues.yaml is when OCNRF Ingress Rate crosses 90 % of 1000 (Maximum ingress request rate)	Major	**description**: 'Total Ingress traffic Rate is above major threshold i.e. 900 requests per second (current value is: {{ $value }})' **summary**: 'timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)'	1.3.6.1.4.1.323.5.3.36.1.2.7002	'oc_ingressgateway_http_requests_total'	The alert is cleared when the total Ingress Traffic rate falls below the Major threshold or when the total traffic rate cross the Critical threshold, in which case the OcnrfTotalIngressTrafficRateAboveCriticalThreshold **Note**: The threshold is configurable in the alerts.yaml alert shall be raised. Steps: Reassess why the OCNRF is receiving additional traffic (for example: geo redundancy OCNRF is unavailable). If this is unexpected, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTotalIngressTrafficRateAboveCriticalThreshold	The total OCNRF Ingress Message rate has crossed the configured critical threshold of 950 TPS. Default value of this alert trigger point in NrfAlertValues.yaml is when OCNRF Ingress Rate crosses 95 % of 1000 (Maximum ingress request rate)	Critical	**description**: 'Total Ingress traffic Rate is above critical threshold i.e. 950 requests per second (current value is: {{ $value }})' **summary**: 'timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: Traffic Rate is above 95 Percent of Max requests per second(1000)'	1.3.6.1.4.1.323.5.3.36.1.2.7003	'oc_ingressgateway_http_requests_total'	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. Note: The threshold is configurable in the alerts.yaml Steps: Reassess why the OCNRF is receiving additional traffic (for example: geo redundancy OCNRF is unavailable). If this is unexpected, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTransactionErrorRateAbove0.1Percent	The number of failed transactions is above 0.1 percent of the total transactions.	Warning	**description**: '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'	1.3.6.1.4.1.323.5.3.36.1.2.7004	'oc_ingressgateway_http_responses_total'	The alert is cleared when the number of failure transactions are below 0.1 percent of the total transactions or when the number of failure transactions cross the 1% threshold in which case the OcnrfTransactionErrorRateAbove1Percent shall be raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. for example: ocnrf_nfDiscover_tx_responses_total with statusCode ~= 2xx. 2. If guidance required, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTransactionErrorRateAbove1Percent	The number of failed transactions is above 1 percent of the total transactions.	Warning	**description**: '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'	1.3.6.1.4.1.323.5.3.36.1.2.7005	'oc_ingressgateway_http_responses_total'	The alert is cleared when the number of failure transactions are below 1% of the total transactions or when the number of failure transactions cross the 10% threshold in which case the OcnrfTransactionErrorRateAbove10Percent shall be raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. for example: ocnrf_nfDiscover_tx_responses_total with statusCode ~= 2xx. 2. If guidance required, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTransactionErrorRateAbove10Percent	The number of failed transactions has crossed the minor threshold of 10 percent of the total transactions.	Minor	**description**: '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'	1.3.6.1.4.1.323.5.3.36.1.2.7006	'oc_ingressgateway_http_responses_total'	The alert is cleared when the number of failure transactions are below 10% of the total transactions or when the number of failure transactions cross the 25% threshold in which case the OcnrfTransactionErrorRateAbove25Percent shall be raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. for example: ocnrf_nfDiscover_tx_responses_total with statusCode ~= 2xx. 2. If guidance required, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTransactionErrorRateAbove25Percent	The number of failed transactions has crossed the minor threshold of 25 percent of the total transactions.	Major	**description**: 'Transaction Error rate is 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'	1.3.6.1.4.1.323.5.3.36.1.2.7007	'oc_ingressgateway_http_responses_total'	The alert is cleared when the number of failure transactions are below 25% of the total transactions or when the number of failure transactions cross the 50% threshold in which case the OcnrfTransactionErrorRateAbove50Percent shall be raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. for example: ocnrf_nfDiscover_tx_responses_total with statusCode ~= 2xx. 2. If guidance required, contact My Oracle Support.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfTransactionErrorRateAbove50Percent	The number of failed transactions has crossed the minor threshold of 50 percent of the total transactions.	Critical	**description**: 'Transaction Error rate is 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'	1.3.6.1.4.1.323.5.3.36.1.2.7008	'oc_ingressgateway_http_responses_total	The alert is cleared when the number of failure transactions are below 50 percent of the total transactions. Steps: 1. Check the Service specific metrics to understand the specific service request errors. for example: ocnrf_nfDiscover_tx_responses_total with statusCode ~= 2xx. 2. If guidance required, contact My Oracle Support.	
OCNRF Application Alerts							

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfRegisteredNFsBelowCriticalThreshold	The number of NFs currently registered with OCNRF is below the critical threshold. Default value of this alert trigger point in NrfAlertValues.yaml is when Registered NFs count with OCNRF is below 2.	Critical	**description**: 'The number of registered NFs detected below critical threshold (current value is: {{ $value }})' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.RequesterNfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: The number of registered NFs detected below	1.3.6.1.4.1.323.5.3.36.1.2.7009	'ocnrf_active_registrations_count'	The alert is cleared when the number of registered NFs are above the critical threshold. Steps: No Action required. This is an information alert.	1. Operator shall configure the threshold values with respect to the number of NFs expected within the network. 2. NFs with NFStatus as 'SUSPENDED' or "UNDISCOVERABLE' shall not be considered as registered.

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			critical threshold.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfRegisteredNFsBelowMajorThreshold	The number of NFs currently registered with OCNRF is below the major threshold. Default value of this alert trigger point in NrfAlertValues.yaml is when Registered NFs count with OCNRF is greater than equal to 2 and less than below 10.	Major	**description**: 'The number of registered NFs detected below major threshold (current value is: {{ $value }})' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: The number of registered NFs detected below major	1.3.6.1.4.1.323.5.3.36.1.2.7010	'ocnrf_active_registrations_count	The alert is cleared when the number of registered NFs are above the major threshold. Steps: No Action required. This is an information alert.	1. Operator shall configure the threshold values with respect to the number of NFs expected within the network. 2. NFs with NFStatus as 'SUSPENDED' or "UNDISCOVERABLE' shall not be considered as registered.

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			threshold. '				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfRegisteredNFsBelowMinorThreshold	The number of NFs currently registered with OCNRF is below the minor threshold. Default value of this alert trigger point in NrfAlertValues.yaml is when Registered NFs count with OCNRF is greater than equal to 10 and less than below 20.	Minor	**description**: 'The number of registered NFs detected below minor threshold (current value is: {{ $value }})' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: The number of registered NFs detected below minor	1.3.6.1.4.1.323.5.3.36.1.2.7011	'ocnrf_active_registrations_count'	The alert is cleared when the number of registered NFs are above the minor threshold. Steps: No Action required. This is an information alert.	1. Operator shall configure the threshold values with respect to the number of NFs expected within the network. 2. NFs with NFStatus as 'SUSPENDED' or "UNDISCOVERABLE' shall not be considered as registered.

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			threshold. '				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfRegisteredNFsBelowThreshold	The number of NFs currently registered with OCNRF is approaching minor threshold. Default value of this alert trigger point in NrfAlertValues.yaml is when Registered NFs count with OCNRF is greater than equal to 20 and less than below 30.	Warning	**description:** 'The number of registered NFs is approaching minor threshold (current value is: {{ $value }})' **summary:**'kubernetes_namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: The number of registered NFs approaching minor threshold.'	1.3.6.1.4.1.323.5.3.36.1.2.7012	'ocnrf_active_registrations_count'	The alert is cleared when the number of registered NFs are approaching minor threshold. Steps: No Action required. This is an information alert.	1. Operator shall configure the threshold values with respect to the number of NFs expected within the network. 2. NFs with NFStatus as 'SUSPENDED' or "UNDISCOVERABLE' shall not be considered as registered.

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfDbReplicationStatusInactive	The db tier replication service status is inactive across the georedundant OCNRFs. The Alarm is raised/ cleared only if the Georedundancy feature is enabled.	Critical	description: 'The Database Replication Status is currently INACTIVE.' summary: 'kubernetes_namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, dbreplicationstatus: {{$labels.DbReplicationStatus}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: The database replication status is INACTIVE.'	1.3.6.1.4.1.323.5.3.36.1.2.7013	'ocnrf_dbreplication_status'	The alert is cleared when the dbtier replication services is active.	The Alarm shall be included only if the Georedundancy feature is enabled.

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfAccessTokenRequestsRejected	OCNRF rejected an AccessToken Request	Warning	**description**: 'AccessToken request(s) have been rejected by OCNRF (current value is: {{ $value }})' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} AccessToken Request has been rejected by OCNRF.'	1.3.6.1.4.1.323.5.3.36.1.2.7014	'ocnrf_accessToken_tx_rejected_total'	The alert is cleared automatically. Steps: The Rejection Reason shall be present in the alert. In case the RejectionReason is AuthScreeningFailed/ ClientNotAuthorized, either the configurations need to be reevaluated or check the consumer NF that has requested for unauthorized token. For other reason, follow the RejectionReason.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfNfAuthenticationFailureRequestsRejected	OCNRF rejected a service request due to NF authentication failure	Warning	**description**: 'Service request(s) received from NF have been rejected by OCNRF (current value is: {{ $value }})' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}} , podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : Request rejected for Nf FQDN based Authentication failure.'	1.3.6.1.4.1.323.5.3.36.1.2.7015	'ocnrf_nf_authentication_failure_total'	The alert is cleared automatically. Steps: No Action required for OCNRF. This is an information alert. The Response Reason shall be present in the alert	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfAccessTokenCurrentKeyIdNotConfigured	OCNRF Access Token Rejected due to CurrentKeyId not configured	Critical	**description**: 'AccessToken request(s) have been rejected by OCNRF (current value is: {{ $value }})' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} AccessToken Request has been rejected by OCNRF as Current Key Id is not	1.3.6.1.4.1.323.5.3.36.1.2.7033	ocnrf_accessToken_tx_rejected_total	The alert is cleared automatically as this will be raised when OCNRF recieves Access Token Request and at that point Current Key Id is not selected.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			configured.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfAccessTokenCurrentKeyIdInvalidDetails	OCNRF Access Token Rejected due to token signign details correspondign to CurrentKeyId are invalid	Critical	**description**: 'AccessToken request(s) have been rejected by OCNRF (current value is: {{ $value }})' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, KeyType: {{$labels.KeyType}}, RejectionReason: {{$labels.RejectionReason}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} AccessToken	1.3.6.1.4.1.323.5.3.36.1.2.7034	ocnrf_accessToken_tx_rejected_total	The alert is cleared automatically as this will be raised when OCNRF receives Access Token Request and at that point Current Key Id details are invalid.	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			Request has been rejected by OCNRF as CurrentKeyId details are invalid.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfOauthCurrentKeyNotConfigured	Oauth Current Key ID is not configured	Critical	**description**: 'OCNRF Oauth Access token Current Key Id is not configured' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} OCNRF Oauth Access token Current Key Id is not configured.'	1.3.6.1.4.1.323.5.3.36.1.2.7035	ocnrf_oauth_currentKeyId_configuredStatus	The alert is cleared when current key id is configured. Steps: Configure valid current key id in Access Token Configuration	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfOauthCurrentKeyDataHealthStatus	Oauth Current Key ID details health is not good	Critical	**description**: 'OCNRF Oauth Access token Current Key Id status is not healthy' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, KeyId: {{$labels.KeyId}}, KeyType: {{$labels.KeyType}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} OCNRF Oauth Access token Current Key Id status is	1.3.6.1.4.1.323.5.3.36.1.2.7036	ocnrf_oauth_keyData_healthStatus	The alert is cleared when current key id status is healthy. Steps: Key Data Health Status details can be checked using OCNRF configuration status REST APIs and configuration microservice logs. Rectify the condition by checking ErrorCondition	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			not healthy.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfOauthNonCurrentKeyDataHealthStatus	Oauth Non Current Key details health is not good	Info	**description**: 'OCNRF Oauth Access token Non current Key Id status is not healthy' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, KeyId: {{$labels.KeyId}}, KeyType: {{$labels.KeyType}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} OCNRF Oauth Access token non current Key Id	1.3.6.1.4.1.323.5.3.36.1.2.7037	ocnrf_oauth_keyData_healthStatus	The alert is cleared when current key id status is healthy. Steps: Key Data Health Status details can be checked using OCNRF configuration status REST APIs and configuration microservice logs. Rectify the condition by checking ErrorCondition	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			status is not healthy.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfOauthCurrentCertificateExpiringIn24Hours	Oauth Current Key ID details are expiring in less than 24 hours	Critical	**description**: 'OCNRF Oauth Access token current Key Id certificate is expiring in less than 24 hours' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, KeyId: {{$labels.KeyId}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} OCNRF Oauth Access token current Key Id certificate is	1.3.6.1.4.1.323.5.3.36.1.2.7038	ocnrf_oauth_keyData_expiryStatus	The alert is cleared when key expiry time is more than 24 hours. Steps: Replace expiring certificate key pair with new ones	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			expiring in less than 24 hours.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfOauthNonCurrentCertificateExpiringIn24Hours	Oauth Non Current Key ID details are expiring in less than 24 hours	Info	**description**: 'OCNRF Oauth Access token non current Key Id certificate is expiring in less than 24 hours' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}} , podname: {{$labels.kubernetes_pod_name}}, KeyId: {{$labels.KeyId}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} OCNRF Oauth Access token non current Key Id certificate is	1.3.6.1.4.1.323.5.3.36.1.2.7039	ocnrf_oauth_keyData_expiryStatus	The alert is cleared when key expiry time is more than 24 hours. Steps: Replace expiring certificate key pair with new ones	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			expiring in less than 24 hours.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfOauthCurrentCertificateExpiringIn30days	Oauth Current Key ID details are expiring in more than 24 hours and less than 30 days	Critical	**description**: 'OCNRF Oauth Access token current Key Id certificate is expiring in less than 30 days' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}} , podname: {{$labels.kubernetes_pod_name}}, KeyId: {{$labels.KeyId}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} OCNRF Oauth Access token current Key Id certificate is	1.3.6.1.4.1.323.5.3.36.1.2.7040	ocnrf_oauth_keyData_expiryStatus	The alert is cleared when certificate for current key id's expiry time is more than 30 days. Steps: Replace expiring certificate key pair with new ones	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			expiring in less than 30 days.'				

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfOauthNonCurrentCertificateExpiringIn30days	Oauth Non Current Key ID details are expiring in more than 24 hours and less than 30 days	Info	**description**: 'OCNRF Oauth Access token non current Key Id certificate is expiring in less than 30 days' **summary**: 'kubernetes_namespace: {{$labels.kubernetes_namespace}},nrflevel: {{$labels.NrfLevel}} , podname: {{$labels.kubernetes_pod_name}}, KeyId: {{$labels.KeyId}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} OCNRF Oauth Access token non current Key Id certificate is	1.3.6.1.4.1.323.5.3.36.1.2.7041	ocnrf_oauth_keyData_expiryStatus	The alert is cleared when certificate for non-current key id's certificate expiry time is more than 30 days. Steps: Replace expiring certificate key pair with new ones	

Table 10-13 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			expiring in less than 30 days.'				

OCNRF Alert Configuration

This section describes the Measurement based Alert rules configuration for OCNRF. The Alert Manager uses the Prometheus measurements values as reported by microservices in conditions under alert rules to trigger alerts.

Note:

- Alert file is packaged with OCNRF custom templates. The OCNRF templates.zip file can be downloaded from MOS. Unzip the OCNRF templates.zip file to get NrfAlertRules.yaml file.
- Review the NrfAlertRules.yaml file and edit the value of the parameters in the NrfAlertRules.yaml file (if needed to be changed from default values) before configuring the alerts. See below table for details.
- kubernetes_namespace is configured as kubernetes namespace in which NRF is deployed. Default value is OCNRF. Please update the NrfAlertRules.yaml file to reflect the correct OCNRF kubernetes namespace.

Alert details which can be updated in NrfAlertRules.yaml file before configuration

Table 10-14 Alerts

Alert Name	Details	Default Value	Notes
OcnrfTotalIngressTrafficRateAboveMinorThreshold	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

Table 10-14 (Cont.) Alerts

Alert Name	Details	Default Value	Notes
OcnrfTotalIngressTrafficRateAboveMajorThreshold	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
OcnrfTotalIngressTrafficRateAboveCriticalThreshold	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

OCNRF Alert configuration in Prometheus

This section describes the measurement based Alert rules configuration for OCNRF in Prometheus.

NAME :- Helm Release of Prometheus

Namespace :- Kubernetes NameSpace in which Prometheus is installed

1. Take Backup of current configuration map of Prometheus:

   ```
   kubectl get configmaps _NAME_-server -o yaml -n _Namespace_ > /tmp/
   tempConfig.yaml
   ```

2. Check and add OCNRF Alert file name inside Prometheus configuration map:

   ```
   sed -i '/etc\/config\/alertsnrf/d' /tmp/tempConfig.yaml
   sed -i '/rule_files:/a\  \- /etc/config/alertsnrf' /tmp/
   tempConfig.yaml
   ```

3. Update configuration map with updated file name of OCNRF alert file:

   ```
   kubectl replace configmap _NAME_-server -f /tmp/tempConfig.yaml
   ```

4. Add OCNRF Alert rules in configuration map under file name of OCNRF alert file:

   ```
   kubectl patch configmap _NAME_-server -n _Namespace_--type merge --
   patch
   "$(cat ~/NrfAlertrules.yaml)"
   ```

> **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 OCNRF Alerts have been reloaded.

Disable OCNRF Alert in Prometheus

Steps to disable Alerts in Prometheus:

1. Edit **NrfAlertrules.yaml** file to remove specific alert:
   Sample alert content from **NrfAlertrules.yaml** is below. This is to provide idea of a specific alert details in **NrfAlertrules.yaml** which need to be disabled.

```
## ALERT SAMPLE START##
      - alert: OcnrfTrafficRateAboveMinorThreshold
        annotations:
          description: 'Ingress traffic Rate is above minor
threshold i.e. 800 mps (current value is: {{ $value }})'
          summary: 'Traffic Rate is above 80 Percent of Max
requests per second(1000)'
        expr:
sum(rate(oc_ingressgateway_http_requests_total{app_kubernetes_io_nam
e="ingressgateway",kubernetes_namespace="ocnrf"}[2m])) >= 800 < 900
        labels:
          severity: Minor
## ALERT SAMPLE END##
```

2. Remove specific alert content which need to be disabled.
3. Perform Alert configuration again. See OCNRF Alert configuration in Prometheus section above for detailed steps.

Disabling Alerts

This section explains the procedure to disable the alerts in OCNRF.

1. Edit **NrfAlertrules.yaml** file to remove specific alert.
2. Remove complete content of the specific alert from the **NrfAlertrules.yaml** file.
   For example: If you want to remove `OcnrfTrafficRateAboveMinorThreshold` alert, remove the complete content:

```
## ALERT SAMPLE START##

      - alert: OcnrfTrafficRateAboveMinorThreshold
        annotations:
          description: 'Ingress traffic Rate is above minor threshold i.e.
800 mps (current value is: {{ $value }})'
          summary: 'Traffic Rate is above 80 Percent of Max requests per
second(1000)'
        expr:
sum(rate(oc_ingressgateway_http_requests_total{app_kubernetes_io_name="ingres
sgateway",kubernetes_namespace="ocnrf"}[2m])) >= 800 < 900
        labels:
          severity: Minor
## ALERT SAMPLE END##
```

3. Perform Alert configuration. See OCNRF Alert Configuration section above for details.

Configuring SNMP Notifier

This section describes the procedure to configuring SNMP Notifier.

Configure and Validate Alerts in Prometheus Server

Refer to OCNRF Alert Configuration section for procedure to configure the alerts.

Validating Alerts
After configuring the alerts in Prometheus server, a user can verify that by following steps:

- Open the Prometheus server from your browser using the <IP>:<Port>
- Navigate to **Status** and then **Rules**
- Search **Ocnrf**. OcnrfAlerts list is displayed.

> **Note:**
>
> If you are unable to see the alerts, it means the alert file is not loaded in a proper format which the Prometheus server accepts. Modify the file and try again.

Configuring SNMP-Notifier
Configure the IP and port of the SNMP trap receiver in the SNMP Notifier using the following procedure:

1. Execute 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
   ```

2. Edit the destination as follows:

   ```
   --snmp.destination=<destination_ip>:<destination_port>
   ```

   Example:

   ```
   --snmp.destination=10.75.203.94:162
   ```

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:

```
2020-04-29 15:34:24 10.75.203.103 [UDP: [10.75.203.103]:2747-
>[172.17.0.4]:162]:DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks:
(158510800) 18 days, 8:18:28.00        SNMPv2-MIB::snmpTrapOID.0
= OID: SNMPv2-SMI::enterprises.323.5.3.36.1.2.7003
SNMPv2-SMI::enterprises.323.5.3.36.1.2.7003.1 =
STRING: "1.3.6.1.4.1.323.5.3.36.1.2.7003[]"  SNMPv2-
SMI::enterprises.323.5.3.36.1.2.7003.2 = STRING: "critical"
SNMPv2-SMI::enterprises.323.5.3.36.1.2.7003.3 = STRING: "Status:
critical- Alert: OcnrfActiveSubscribersBelowCriticalThreshold  Summary:
namespace: ocnrf, nftype:5G_EIR, nrflevel:6faf1bbc-6e4a-4454-a507-
a14ef8e1bc5c, podname: ocnrf-nrfauditor-6b459f5db5-4kvt4,
        timestamp: 2020-04-29 15:33:24.408 +0000 UTC: Current number
of registered NFs detected below critical threshold.  Description: The
number of registered NFs detected below critical threshold (current
value
          is: 0)
```

MIB Files for OCNRF

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.

- OCNRF-MIB-TC-1.10.0.mib
  This is considered as OCNRF top level mib file, where the Objects and their data types are defined.
- OCNRF-MIB-1.10.0.mib
  This file fetches the Objects from the top level mib file and based on the Alert notification, these objects can be selected for display.

> **Note:**
>
> MIB files are packaged along with OCNRF Custom Templates. Download the file from MOS. Refer to OCNRF Installation and Upgrade guide for more details.