Oracle® Communications

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

Release 1.8.0
F35154-03
November 2020

Oracle Communications Network Repository Function (NRF) Cloud Native User's Guide, Release 1.8.0

F35154-03

Contents

6 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 1.8.x in Oracle Communications Cloud Native Network Repository Function (NRF) User's Guide.

New and Updated Features in Release 1.8.0

- Updated the KPI section.
- Added **Access Token Request Authorization feature** and **NF Authentication using TLS certificate feature** parameters in General Configurations.
- Added **Logging Level** Configuration parameters in General Configurations.
- Added Configuring Access Token Request Authorization feature details.
- Added Configuring NF Authentication using TLS certificate feature details.
- Added new alerts and resolutions in OCNRF Alerts section.
- Added new metrics to 1.8.0 features in OCNRF Metrics section.

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 & Load info 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 notifications to subscribed NFs.
- **Retrival 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.

OCNRF Features

This section explains the OCNRF features.

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.

For configuring NF Screening feature, see Configuring NF Screening.

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.

Subscriber Location Function

OCNRF supports Subscriber Location Function (SLF) feature which identifies specific NF Type selection based on subscriber identity. For NF selection based on subscriber identity, OCNRF performs the following:

- Identifies (if received) NFDiscover service request requires NF selection based on subscriber identity.
- Discovers the NF Group Id(s) using Nudr_GroupIDmap (aka SLF) Query service operation.
- Generates NFDiscover service response using NF Group Id(s) and other parameters.

OCNRF Forwarding Feature

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

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

An 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).

OCNRF Geo-Redundancy Feature

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 replicate 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:

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

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.

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 shall keep its operative status alive by sending NF Heart-Beat 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 shall 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 shall 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 Heart-beat for the heartbeat interval, NRF shall mark the NFProfile as SUSPENDED. The NFProfile and its services shall not longer be discoverable by the other NFs through the NfDiscovery Service. Further, the NRF shall 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.

OCNRF Access Token Request Authorization Feature

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

OCNRF 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 shall obtain an OAuth2 access token from the NRF. NRF shall perform the required authorization, and if successful shall issue the token with the requested claims. Using this feature, OCNRF provides the user an option to tailor the authorization of the Producer-Consumer NF Types along with the Producer NF's services.

User can configure mapping of the RequesterNfType, TargetNfType and the allowedServices 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 NRF when an Access Token request is rejected.

Access Token configurable attribute "logicalOperatorForScope" will be 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 shall be present in the allowed Services. If it is set to "AND", all the services in the scope shall be present in the allowed services.

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.

4
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.
- ocnrfEndPointHost: OCNRF EndPoint Host's FQDN.
- ocnrfEndPointPort: OCNRF EndPoint Host's Port.

OCNRF Host Configuration

OCNRF's NfHostConfig Configuration attribute allows to configure the details of NRF and SLF/UDR Network Functions. These attributes (nrfHostConfig and slfHostConfig) used for NRF forwarding and Subscriber Location Function (SLF) features respectively.

The NfHostConfig 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 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 nfHostConfig 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.

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 nfHostConfig 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.

General Configurations

The section provides information for performing general configurations in OCNRF.

General configuration - OCNRF system options

Table 4-1 Service API Interface

Resource Name	Resource URI	HTTP Method or Custom Operation	Description
nrf-configuration (Store)	{apiRoot}/nrf-configuration/v1/system-options	GET	Retrieves OCNRF system options configuration
nrf-configuration (Store)	{apiRoot}/nrf-configuration/v1/system-options	PUT	Updates OCNRF system options configuration

Resource Standard Methods
GET - Retrieve OCNRF System options configuration

Table 4-2 Data structures supported by the GET Response Body

Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Cardinality	Response Codes	Description
ProblemDetails	M	1	500 Internal Server Error	The response body contains the error reason of the request message.
NrfSystemOptions	M	1	200 OK	Response body contains the OCNRF current system options

PUT - Update OCNRF System options configuration

Table 4-3 Data structures supported by the PUT Request Body

Data Type	P	Cardinality	Description
NA	M	1	NrfSystemOptions details

Table 4-4 Data structures supported by the PUT Response Body

Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Cardinality	Response Codes	Description
ProblemDetails	M	1	500 Internal Server Error	The response body contains the error reason of the request message.
ProblemDetails	M	1	400 Bad request	The response body contains the error reason of the request message.
NrfSystemOptions	M	1	200 OK	Specifies that the update of NrfSystemOptions is successful and provides the values in database.

REST Message Sample

Request_Type: GET and PUT

URL
: *http://<k8s host>:<port>/nrf-configuration/v1/system-options*

```
{
    "generalSystemOptions": {
        "nrfPlmnList": [{
            "mcc": "310",
            "mnc": "14"
        }],
        "enableF3": true,
```

```
        "enableF5": true,
        "maximumHopCount": 3,
        "defaultLoad": 5,
        "defaultPriority": 100,
        "addPriorityInNFProfile": false,
        "addLoadInNFProfile": false,
        "ocnrfEndPointHost": "ocnrf-
ingressgateway.ocnrf.svc.cluster.local",
        "ocnrfEndPointPort": 80
    },
    "nfScreeningSystemOptions": {
        "nfScreeningFeatureStatus": "DISABLED",
        "nfScreeningFailureHttpCode": 403
    },
    "nfAccessTokenSystemOptions": {
        "oauthTokenAlgorithm": "ES256",
        "oauthTokenExpiryTime": "1h",
        "authorizeRequesterNf": "ENABLED",
        "logicalOperatorForScope": "AND",
        "audienceType": "NF_INSTANCE_ID",
        "authFeatureConfig":{
           "authFeatureStatus":"DISABLED",
           "authConfig":[{
              "targetNfType":"AMF",
              "requesterNfType":"UDM",
              "serviceNames":["namf-loc"]
           }],
           "authErrorResponses":[{
              "errorCondition":"RequesterNf_Unauthorized",
              "errorCode":400,
              "errorResponse":"The Consumer NfType is not authorized to
receive access token for the requested Nftype."
           }]
        }
    },
    "nfManagementSystemOptions": {
        "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,
        "subscriptionValidityDuration": "24h",
```

```
            "nrfSupportForProfileChangesInNotification": false,
            "nfProfileSuspendDuration": "168h",
            "acceptAdditionalAttributes": false,
            "allowDuplicateSubscriptions": true
        },
        "nfDiscoverSystemOptions": {
            "discoveryValidityPeriod": "1h",
            "profilesCountInDiscoveryResponse": 3,
            "discoveryResultLoadThreshold": 0
        },
        "slfSystemOptions": {
            "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]
            },
            "slfFeatureStatus": "DISABLED"
        },
        "nfAuthenticationSystemOptions": {
            "nfRegistrationAuthenticationStatus": "DISABLED",
            "nfSubscriptionAuthenticationStatus": "DISABLED",
            "nfDiscoveryAuthenticationStatus": "DISABLED",
            "accessTokenAuthenticationStatus": "DISABLED",
            "nfProfileRetrievalAuthenticationStatus": "DISABLED",
            "nfListRetrievalAuthenticationStatus": "DISABLED",
            "checkIfNfIsRegistered": "DISABLED",
            "nfAuthenticationErrorResponses": [{
                    "errorCondition": "Nf_Fqdn_Authentication_Failure",
                    "errorCode": 403,
                    "errorResponse": "Failed to authenticate NF using FQDN",
                    "retryAfter": "5m"}]
        },
        "errorResponses": {
            "slfErrorResponses": [{
                "errorCondition": "SLF_Missing_Mandatory_Parameters",
                "errorCode": 400,
                "errorResponse": "Mandatory parameter missing for SLF
    Lookup"
            }, {
                "errorCondition": "SLF_GroupId_NotFound",
                "errorCode": 404,
                "errorResponse": "Group Id Not found from SLF"
            }, {
                "errorCondition": "SLF_Not_Reachable",
```

```
                "errorCode": 504,
                "errorResponse": "SLF not reachable"
            }],
            "nrfForwardingErrorResponses": [{
                "errorCondition": "NRF_Not_Reachable",
                "errorCode": 504,
                "errorResponse": "NRF not reachable"
            }, {
                "errorCondition": "NRF_Forwarding_Loop_Detection",
                "errorCode": 508,
                "errorResponse": "Loop Detected"
            }]
          },
        "forwardingSystemOptions": {
            "profileRetreivalForwardingStatus": "DISABLED",
            "subscriptionForwardingStatus": "DISABLED",
            "discoveryForwardingStatus": "DISABLED",
            "accessTokenForwardingStatus": "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}$"
            }
        },
        "geoRedundancySystemOptions": {
            "geoRedundancyFeatureStatus": "DISABLED",
            "replicationLatency": "5s",
            "monitorNrfServiceStatusInterval": "5s",
            "monitorDBReplicationStatusInterval": "5s"
        },
        "loggingLevelSystemOptions": {
            "nfSubscriptionLogLevel": "WARN",
            "nfRegistrationLogLevel": "WARN",
            "nfDiscoveryLogLevel": "WARN",
            "nfAccessTokenLogLevel": "WARN",
            "nrfAuditorLogLevel": "WARN",
            "nrfConfigurationLogLevel": "WARN",
     }
    }
```

Data Model

> **Note:**
>
> At least one attribute must be present to ensure that the PUT request is not empty.

Table 4-5 NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
generalSystemOptions	nrfPlmnList	array (PlmnId)			This value will have at least one PLMN supported by OCNRF and this value is set before using OCNRF. See the footnote.
generalSystemOptions	enableF3	ENUM (true or false)	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.
generalSystemOptions	enableF5	ENUM (true or false)	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.
generalSystemOptions	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.
generalSystemOptions	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.
generalSystemOptions	addLoadInNFProfile	ENUM (true or false)	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.
generalSystemOptions	addPriorityInNFProfile	ENUM (true or false)	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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
generalSystemOptions	maximumHopCount	INTEGER	1-5	3	Maximum number of Nodes (SLF/NRF's) that OCNRF can communicate, to service a request.
generalSystemOptions	ocnrfEndPointHost	STRING	None	ocnrf-ingressgateway.ocnrf.svc.cluster.local	ocnrfEndPointHost needs to be OCNRF's External Routable FQDN (e.g. ocnrf.oracle.com) OR External Routable IpAddress (e.g. 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 NfListRetrival Service Operation response.
generalSystemOptions	ocnrfEndPointPort	INTEGER	None	80	OCNRF EndPoint Host's Port

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
forwardingSystemOptions	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 2 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. See the footnote.
forwardingSystemOptions	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 nfHostConfig for details for Primary and Secondary NRF details. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
forwardingSystemOptions	profileRetrievalForwardingStatus	String (Feature Status)		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 NfProfileRetrival 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 NfProfileRetrival request in any case. It will return a response to consumer NF without forwarding it. See the footnote. See the footnote.
forwardingSystemOptions	subscriptionForwardingStatus	String (Feature Status)		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 the NfInstanceIdCond condition is requested in the Subscription Request. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
forwardingSystemOptions	discoveryForwardingStatus	String (Feature Status)		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. See the footnote.
forwardingSystemOptions	accessTokenForwardingStatus	String (Feature Status)		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. See the footnote.
nfScreeningSystemOptions	nfScreeningFeatureStatus	String (Feature Status)		DISABLED	This attribute indicates if NF Screening Feature is enabled or not. See the footnote.
nfScreeningSystemOptions	nfScreeningFailureHttpCode	INTEGER		403	This attribute will inform what HTTP status code will be returned if incoming request does not pass NF Screening rules barrier. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
nfManagementSystemOptions	nfHeartbeatTimers	array (HeartbeatInfo)			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.
nfManagementSystemOptions	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. See the footnote.
nfManagementSystemOptions	nrfSupportForProfileChangesInResponse	ENUM (true or false)	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. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
nfManagementSystemOptions	subscriptionValidityDuration	String	10s - 720h	24h	If Validity time attribute is not received in SubscriptionData during NFSubscribe, this default value will be used for calculation of validity time (current time + default duration). If Validity time attribute is received in SubscriptionData during NFSubscribe, this is minimum value will be used for validation and limit purpose. It means if value provided is less than (current time + minimum value), then calculated value with minimum duration value will be considered as validity time of subscription and similarly in case validity time is more than (current time + maximum duration), then calculated value with maximum duration will be considered as validity time of 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. See the footnote.
nfManagementSystemOptions	nrfSupportForProfileChangesInNotification	ENUM (true, false)	true or false	false	OCNRF sends profileChanges attribute instead of NFProfile in Notification, if this flag is enabled. See the footnote.
nfManagementSystemOptions	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 & seconds respectively. See the footnote.
nfManagementSystemOptions	acceptAdditionalAttributes	ENUM (true, false)	true or false	false	OCNRF preserves additional attributes that are not defined **by 3GPP** in NFProfile/ NFService in the database based on this attribute value. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
nfManagementSystemOptions	allowDuplicateSubscriptions	ENUM (true, false)	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 NFSubscribe service operation.
nfDiscoverSystemOptions	discoveryValidityPeriod	String	1s - 168h	1h	This attribute mentions the validity period of a discovery request 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. See the footnote.
nfDiscoverSystemOptions	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 will get disabled, in that case all the profiles will be returned. If GET option returns this attribute value as 0, then it means this feature is disabled. **Note**:- If Limit attribute is present in SearchData URI then this attribute is not used.
nfDiscoverSystemOptions	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.
nfAccessTokenSystemOptions	oauthTokenAlgorithm	String (oauthTokenAlgorithm)		ES256	Access token key algorithm which will be used to sign the oauth token. See the footnote.
nfAccessTokenSystemOptions	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 & seconds respectively. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
nfAccessTokenSystemOptions	authorizeRequesterNf	String (Feature Status)		ENABLED	This attribute validates the requester NF is registered with OCNRF or not. OCNRF issues the access token only to the registered requester NFs. If the value is Disabled, OCNRF will issue token to non-registered NFs as well.
nfAccessTokenSystemOptions	audienceType	String (AudienceType)		NF_INSTANCE_ID	This value decides the AudienceType in AccessTokenClaim. OCNRF considers this value only if targetnfInstanceId is **not received** in AccessTokenRequest.
nfAccessTokenSystemOptions	logicalOperatorForScope	String (LogicalOperatorForScope)		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.
nfAccessTokenSystemOptions	authFeatureConfig	AuthFeatureConfig			The attribute contains the parameters required to enable and configure NfAccessToken Authorization Feature.
slfSystemOptions	slfFeatureStatus	String (Feature Status)		DISABLED	Enables/disables the SLF Feature. See NOTE 1.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
slfSystemOptions	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. See the footnote.
slfSystemOptions	supportedNfTypeList	array			NF Type list for which SLF need to be supported. SLF look up will happen only for NF Types mentioned in this configuration. To reset this attribute, send empty array, for example:-"supportedNfTypeList" : [ ] If this value is set, then slfHostConfig is also set. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
slfSystemOptions	preferredSubscriberIdType	String (SubscriberIdType)	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 is used for the query to SLF. See the footnote.
slfSystemOptions	rerouteOnResponseHttpStatusCodes	String (ResponseHttpStatusCodes)		"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). See the footnote.
geoRedundancySystemOptions	geoRedundancyFeatureStatus	String (Feature Status)		DISABLED	Enables/Disables the geoRedundancy feature in OCNRF. See the footnote.
geoRedundancySystemOptions	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.
geoRedundancySystemOptions	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.
geoRedundancySystemOptions	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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
errorResponses	slfErrorResponses	array (ErrorInfo)			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. See the footnote.
errorResponses	nrfForwardingErrorResponses	array (ErrorInfo)			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. See the footnote.
nfAuthenticationSystemOptions	nfAuthenticationErrorResponses	array (ErrorInfo)			This attribute defines the error responses which may be sent for NF Authentication scenarios. This attribute will allow to update the error response code, error response description,retryAfter and redirectUrl for preloaded error condition. See the footnote.
nfAuthenticationSystemOptions	nfRegistrationAuthenticationStatus	String (Feature Status)		DISABLED	This attribute controls the authentication of consumer NF for NfRegister, NfUpdate and NfDeregister service operations. If this attribute is enabled then identity of consumer NF is validated. If this attribute is disabled then validation is not performed for consumer NF.
nfAuthenticationSystemOptions	nfSubscriptionAuthenticationStatus	String(Feature Status)		DISABLED	This attribute controls the authentication of consumer NF for NfStatusSubscribe and NfStatusUnsubscribe service operations. If this attribute is enabled then identity of consumer NF is validated and NRF allows the subscription only if the NF is registered with NRF. If this attribute is disabled then validation is not performed for consumer NF.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
nfAuthenticationSystemOptions	nfDiscoveryAuthenticationStatus	String (Feature Status)		DISABLED	This attribute controls the authentication of consumer NF for NfDiscover service operations. If this attribute is enabled then NF identity of consumer NF is validated. If 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.
nfAuthenticationSystemOptions	accessTokenAuthenticationStatus	String (Feature Status)		DISABLED	This attribute controls the authentication of consumer NF for AccessToken service operation. If this attribute is enabled then identity of consumer NF is validated. If this attribute is disabled then validation is not performed for consumer NF.
nfAuthenticationSystemOptions	nfProfileRetrivalAuthenticationStatus	String(Feature Status)		DISABLED	This attribute controls the authentication of consumer NF for NfProfileRetrieval service operation. If this attribute is enabled then NF identity is validated against registered NF Profiles. If this attribute is disabled then validation is not performed for consumer NF.
nfAuthenticationSystemOptions	nfListRetrievalAuthenticationStatus	String (Feature Status)		DISABLED	This attribute controls the authentication of consumer NF for NfListRetrieval service operation. If this attribute is enabled then NF identity is validated against registered NF Profiles. If this attribute is disabled then validation is not performed for consumer NF.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	Default Values	Description
nfAuthenticationSystemOptions	checkIfNfIsRegistered	String (Feature Status)		DISABLED	This attribute controls the authentication of consumer identity against the registered profiles in database. If this attribute is enabled then for below mentioned case NF identity of registered profiles in database is validated: • discovery request does not contain requester-nf-instance-fqdn and nfDiscoveryAuthenticationStatus is enabled. If this attribute is disabled then validation is not performed for consumer NF.
loggingLevelSystemOptions	nfSubscriptionLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Logging Level for the NFSubscription Microservice
loggingLevelSystemOptions	nfRegistrationLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Logging Level for the NFRegistration Microservice
loggingLevelSystemOptions	nfDiscoveryLogLevel	string	OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE	WARN	Logging Level for the NFdiscovery Microservice

Table 4-5 (Cont.) NrfSystemOptions - Parameters

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

Note:

If the attribute is not present, existing value in database is used. It can be the default value or the last updated value. But at least one attribute must be present so that the PUT request is not empty.

Table 4-6 General Data Types

Data Type	Reference
NFType	3GPP TS 29.510
NFServiceVersion	3GPP TS 29.510
UriScheme	3GPP TS 29.510
Fqdn	3GPP TS 29.510

Table 4-7 Feature Status

Enumeration value	Description
ENABLED	Enables the feature.
DISABLED	Disables the feature.

Table 4-8 OauthTokenAlgorithm

Enumeration value	Description
ES256	ES256 algorithm key will be used to sign the oauth token
RS256	RS256 algorithm key will be used to sign the oauth token

Table 4-9 AudienceType

Enumeration value	Description
NF_INSTANCE_ID	NF Instance Id(s) in audience IE of AccessTokenClaim.
NF_TYPE	NF Type in audience IE of AccessTokenClaim.

Table 4-10 LogicalOperatorForScope

Enumeration value	Description
AND	If value is AND, while looking for profiles of producer network function, OCNRF issues token for all profiles matching with services-names present in the scope.
OR	If value is OR, OCNRF includes producers matching with any of the services-names present in scope, while looking for profiles of producer NFs.

Table 4-11 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 its HTTPS
apiPrefix	string	O	ApiPrefix
priority	integer	M	Priority of NF
nfInstanceId	string	M	nfInstanceId of NF

Table 4-12 SubscriberIdType

Enumeration Value	Description
SUPI	Subscriber Id is SUPI
GPSI	Subscriber Id is GPSI

Table 4-13 ErrorInfo

Attribute	DataType	Presence	Description
errorCondition	ErrorCondition	ReadOnly	Error Conditions
errorCode	Integer	M	This response code will be used when corresponding error condition will occur.
errorResponse	String	M	This response description will be used when corresponding error condition will occur.
retryAfter	Duration	O	The attribute indicates the time interval after which the NF retries the request. the attribute is included in retryAfter header of Error Response by the OCNRF only where error_response_code is present in retryAfterErrorCodes list introduced in general engineering system options. 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. No validation will be performed on retryAfter attribute. Configuration will be accepted for retryAfter attribute even when its not in confluence with error_response_code being configured. Range: 60s-1h Default Value: 5m
redirectUrl	String	O	The attribute indicates the NF to redirect its request to this uri. the attribute is included in location header of Error Response by the OCNRF only where error_response_code is present in redirectUrlErrorCodes list present in general engineering system options. redirectUrl should be in URI format. Its mandatory to configure redirectUrl when error_response_code configured is present in redirectUrlErrorCodes list introduced in general engineering system options.

Table 4-14 ErrorCondition

Error Condition	Error Response Code	Description
SLF_Missing_Mandatory_Parameters	400	SLF mandatory parameters are missing
SLF_Not_Reachable	504	SLF is not reachable from OCNRF
SLF_GroupId_NotFound	404	Group Id Not found from SLF
NRF_Not_Reachable	504	Primary/Secondary NRF is not reachable from NRF
NRF_Forwarding_Loop_Detection	508	Loop detected while processing NRF Service Operation Message
RequesterNF_Unauthorized	400	The RequesterNfType is not authorized to receive access token for the targetNfType.
Nf_Fqdn_Authentication_Failure	403	Failed to authenticate NF using FQDN

Table 4-15 ResponseHttpStatusCodes

Attribute	DataType	Presence	Description
pattern	String	C	Either pattern or codeList is present.
codeList	array (integer)	C	Either pattern or codeList is present.

Table 4-16 HeartBeatInfo

Attribute	DataType	Presence	Description

nfType	String	M	All nftypes supported in 29.510 Rel 15.5.0. In addition to this, *ALL_NF_TYPE* and *CUSTOM_NF_TYPE* is also supported.*ALL_NF_TYPE* is the NfType to be used to specify the default configuration that is to be used when nfType specific configuration is not present. **Note**: ALL_NF_TYPE is preloaded and cannot be removed. *CUSTOM_NF_TYPE* is the Nftype to be used to specify the configuration for custom nftypes.
			By default record will pre-loaded for ALL_NF_TYPE with values
			
			`"nfHeartbeatTimers": [`
			
			`{`
			
			`"defaultHbTimer": "30s",`
			
			`"maxHbTimer": "5m",`
			
			`"minHbTimer": "30s",`
			
			`"nfHeartbeatMissAllowed": 3,`
			
			`"nfType": "ALL_NF_TYPE"`
			
			`}]`

minHbTimer	Duration	M	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	Duration	M	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.
defaultHbTimer	Duration	M	The default HeartTimer to be used when the NF does not provide. 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	M	The allowed number of missed HeartBeat after which the NFProfile is marked as suspended.

Table 4-17 AuthFeatureConfig

Attribute	DataType	Presence	Description
authFeatureStatus	String (Feature Status)	O	Enables/Disables the NfAccessToken Authorization Feature.
authConfig	array (AuthConfig)	O	The attribute defines the 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.
authErrorResponses	array (ErrorInfo)	O	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 is preloaded. Refer Note.

Note:

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

Table 4-18 AuthConfig

Attribute	DataType	Presence	Description
targetNfType	String	M	The attribute defines the nftype of the target NF.
requesterNfType	String	M	The attribute defines the nftype of the requester NF that is authorized to access the target Nf Type and its services.
serviceNames	array (String)	M	This attribute defines the NF services that is authorized to be accessed by the requester NF type. The value "*" indicates that all the services are authorized to be accessed the requester Nf Type. If "*" is to be used, the services contain only a single entry in the list with this value.

Configuring NF Screening

This section provides information for configuring NF Screening.

Table 4-19 Resources and Methods Overview

Resource Name	Resource URI	HTTP Method or Custom Operation	Description
screening-rules (Store)	{apiRoot}/nrf-configuration/v1/screening-rules	GET	Returns all the screening rules
screening-rules (Document)	{apiRoot}/nrf-configuration/v1/screening-rules/{nfScreeningRulesListType}	GET	Returns screening rules corresponding to the specified NF Screening Rule List Type.
screening-rules (Document)	{apiRoot}/nrf-configuration/v1/screening-rules/{nfScreeningRulesListType}	PUT	Replace the complete specified NF Screening Rule List Type
screening-rules (Document)	{apiRoot}/nrf-configuration/v1/screening-rules/{nfScreeningRulesListType}	PATCH	Partially updates the specified NF Screening Rule List Type.

Resource Standard Methods
PUT - Updates a particular screening rule (except read only attributes)

Table 4-20 Data structures supported by the PUT Request Body

Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Cardinality	Description
NfScreening Rules	M	1	NF Screening Rules which need to be updated.

Table 4-21 Data structures supported by the PUT Response Body

Data Type	Mandatory(M) / Optional(O)/ Conditional(C)	Cardinality	Response Codes	Description
NfScreeningRules			200 OK	Successful response
ProblemDetails	C	1	404 NOT FOUND 500 INTERNAL ERROR 400 BAD REQUEST	The response body contains the error reason of the request message.

PATCH - Updates partially a particular screening rule (except read only attributes)

Table 4-22 Data structures supported by the PATCH Request Body

Data Type	Mandatory(M)/ Optional(O) / Conditional(C)	Cardinality	Description
PatchDocument	M	1	It contains the list of changes to be made to the NF Screening Rule, according to the JSON PATCH format specified in IETF RFC 6902 [13].

Table 4-23 Data structures supported by the PATCH Response Body

Data Type	Mandatory(M)/ Optional(O) / Conditional(C)	Cardinality	Response Codes	Description
NfScreeningRules			200 OK	Successful response
ProblemDetails	C	1	404 NOT FOUND 500 INTERNAL ERROR 400 BAD REQUEST	The response body contains the error reason of the request message.

GET - Collection of screening rules

Table 4-24 URI query parameters supported by the GET method

Name	Data Type	Mandatory(M)/ Optional(O) / Conditional(C)	Cardinality	Description
nfScreeningRulesListType	NfScreeningRulesListType	O	0.1	The type of NF screening rules on this basis of rules list type.
nfScreeningRulesListStatus	NfScreeningRulesListStatus	O	0.1	Screening Rules List on the basis of status (Enabled or Disabled)

Table 4-25 Data structures supported by the GET Response Body

Data Type	Mandatory(M)/ Optional(O) / Conditional(C)	Cardinality	Response Codes	Description
ScreeningRulesResult	M	1	200 OK	The response body contains a list of screening lists, or an empty object if there are no screening rules to return in the query result.
ProblemDetails	C	1	500 INTERNAL ERROR 400 BAD REQUEST	The response body contains the error reason of the request message.

Table 4-26 ScreeningRulesResult - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O) / Conditional(C)	Cardinality	Description
nfScreeningRulesList	array (NfScreeningRules)	M	0.N	It shall contain an array of NF Screening List. An empty array means there is no NF Screening list configured.

GET - Particular screening list rule

Table 4-27 Data structures supported by the GET Response Body

Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Cardinality	Response Codes	Description
NfScreeningRules	M	1	200 OK	The response body contains requested screening list.
ProblemDetails	C	1	500 INTERNAL ERROR 400 BAD REQUEST	The response body contains the error reason of the request message.

REST message samples

Screening List Update

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

URL: *http://host:port/nrf-configuration/v1/ screening-rules /CALLBACK_URI*
Request_Type: PUT

Content-Type: *application/json*
Request Body

NF screening rules to get all of the configured rules

```
{
    "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]
            }
        ]
    }
}
```

URL: *http://host:port/nrf-configuration/v1/ screening-rules /*
Request_Type: 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"
                ]
            }
        }
    ]
}
```

NF screening rules to get a particular configured rule

URL: *http://host:port/nrf-configuration/v1/ screening-rules /CALLBACK_URI*
Request_Type: 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 for partial rule update

http://host:port/nrf-configuration/v1/screening-rules/CALLBACK_URI
Request_Type: PATCH

Content-Type: *application/json-patch+json*
Request Body

```
[
    {"op":"remove","path":"/globalScreeningRulesData/
nfCallBackUriList/2/ports/0"},
    {"op":"replace","path":"/globalScreeningRulesData/
failureAction","value": "CONTINUE"}
]
```

URL: *http://host:port/nrf-configuration/v1/ screening-rules /CALLBACK_URI*
Request_Type: PATCH

Content-Type: *application/json-patch+json*
Response Body

```
[{"op":"add","path":"/nrfScreeningRulesData","value": {"failureAction":
"SEND_ERROR","nfCallBackUriList": [{"ipv4AddressRange":{"start" :
"189.163.192.10","end": "190.178.127.10"}}]}}]
```

Table 4-28 NfScreeningRules - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
nfScreeningRulesListType	NfScreeningRulesListType	C	ReadOnly. It will be returned while retrieving the rule.

Table 4-28 (Cont.) NfScreeningRules - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
nfScreeningType	NfScreeningType	M	Screening type of complete screening list. Blacklist or whitelist. All the rules can be either blacklist or whitelist.
nfScreeningRulesListStatus	NfScreeningRulesListStatus	M	This attribute will enable or disable 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.
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.

Table 4-28 (Cont.) NfScreeningRules - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
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 forNWDAF need to be configured.

Table 4-29 NfScreeningRulesData - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
failureAction	FailureAction	M	Indicates what action needs to be taken during failure.
nfFqdn	NfFqdn	C	If this attribute is present in message it shouldn't be null. This attribute will be present if screeningListType is NF_FQDN.
nfCallBackUriList	array(NfCallBackUri)	C	If this attribute is present in message it shouldn't be null. This attribute will be present if screeningListType is CALLBACK_URI.
nfIpEndPointList	array(NfIpEndPoint)	C	If this attribute is present in message it shouldn't be null. This attribute may be present if screeningListType is NF_IP_ENDPOINT.
plmnList	array(PlmnId)	C	If this attribute is present in message it shouldn't be null. This attribute may be present if screeningListType is PLMN_ID.
nfTypeList	array(NfTypeList)	C	If this attribute is present in message it shouldn't be null. This attribute may be present if screeningListType is NF_TYPE_REGISTER.

Table 4-30 NfScreeningRulesListType - Parameters

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 4-31 NfScreeningType - Parameters

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 4-32 NfScreeningRulesListStatus - Parameters

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

Table 4-33 FailureAction - Parameters

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

Table 4-34 NfFqdn - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
fqdn	array(FQDN)	C	Exact FQDN to be matched. This is conditional, at least one attribute shall be present.
pattern	array(string)	C	Regular Expression for FQDN. This is conditional, at least one attribute shall be present.

Table 4-35 NfIpEndPoint - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ipv4Address	Ipv4Addr	C	IPv4 address to be matched.
ipv4AddressRange	Ipv4AddressRange	C	Range of IPv4 addresses.
ipv6Address	Ipv6Addr	C	IPv6 address to be matched.
ipv6AddressRange	Ipv6AddressRange	C	Range of IPv6 addresses.
port	array(integer)	O	If this attribute is not configured then it will not be considered for validation.

Table 4-35 (Cont.) NfIpEndPoint - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
portRange	array(PortRange)	O	If this attribute is not configured then it will not be considered for validation.

> **Note:**
>
> Depending on the conditions, only one of the ipv4Address, ipv4AddressRange, ipv6Address, and ipv6AddressRange attributes can be present.

Table 4-36 NfCallBackUri - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
fqdn	FQDN	C	Exact Fqdn to be matched.
pattern	string	C	Regular Expression for FQDN, Ipv4Address or Ipv6Address.
ipv4Address	Ipv4Addr	C	IPv4 address to be matched.
ipv4AddressRange	Ipv4AddressRange	C	Range of IPv4 addresses.
ipv6Address	Ipv6Addr	C	IPv6 address to be matched.
ipv6AddressRange	Ipv6AddressRange	C	Range of IPv6 addresses.
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:**
>
> Depending on the conditions, only one of the fqdn, pattern, ipv4Address, ipv4AddressRange, ipv6Address, and ipv6AddressRange attributes can be present.

Table 4-37 PortRange - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
start	integer	M	First value identifying the start of port range.
end	integer	M	Last value identifying the end of port range.

Table 4-38 Ipv6AddressRange - Parameters

Attribute Name	Data type	Mandatory(M)/ Optional(O)/ Conditional(C)	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.

Table 4-39 Common data types

Data Type	Reference
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
ProblemDetails	3GPP TS 29.571

Configuring Access Token Request Authorization

OCNRF 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 shall obtain an OAuth2 access token from the OCNRF. OCNRF shall perform the required authorization, and if successful will issue the token with the requested claims. Using this feature, OCNRF provides the user an option to tailor the authorization of the Producer-Consumer NF Types along with the Producer NF's services.

User can configure mapping of the RequesterNfType, TargetNfType and the allowedServices 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 NRF 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.

Configuration for OCNRF Access Token Request Authorization Feature

Under `nfAccessTokenSystemOptions` parent attribute, `authFeatureConfig` attribute provides the attributes required to use OCNRF Access Token Request Authorization Feature. Refer to General Configurations table for more details.

Sample configuration to use the feature

```
"nfAccessTokenSystemOptions":{
      "oauthTokenAlgorithm":"ES256",
      "oauthTokenExpiryTime":"1h",
      "authorizeRequesterNf":"ENABLED",
      "logicalOperatorForScope":"AND",
      "audienceType":"NF_INSTANCE_ID",
      "authFeatureConfig":{
         "authFeatureStatus":"ENABLED",
         "authConfig":[
            {
               "targetNfType":"PCF",
               "requesterNfType":"AMF",
               "serviceNames":[
                  "npcf-am-policy-control",
                  "npcf-eventexposure"
               ]
            },
            {
               "targetNfType":"UDM",
               "requesterNfType":"AMF",
               "serviceNames":[
                  "*"
               ]
            }
         ],
         "authErrorResponses":[
            {
               "errorCondition":"RequesterNf_Unauthorized",
               "errorCode":400,
               "errorResponse":"The Consumer NfType is not authorized
to receive access token for the requested Nftype."
            }
         ]
      }
   }
```

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 General Configurations table for more details.

Sample configuration to use the feature

Enable the following attributes:

```
"nfAuthenticationSystemOptions": {
       "nfRegistrationAuthenticationStatus": "DISABLED",
       "nfSubscriptionAuthenticationStatus": "DISABLED",
       "nfDiscoveryAuthenticationStatus": "DISABLED",
       "accessTokenAuthenticationStatus": "DISABLED",
       "nfProfileRetrievalAuthenticationStatus": "DISABLED",
       "nfListRetrievalAuthenticationStatus": "DISABLED",
       "checkIfNfIsRegistered": "DISABLED"
   }
```

5
Configuring OCNRF using CNC Console

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

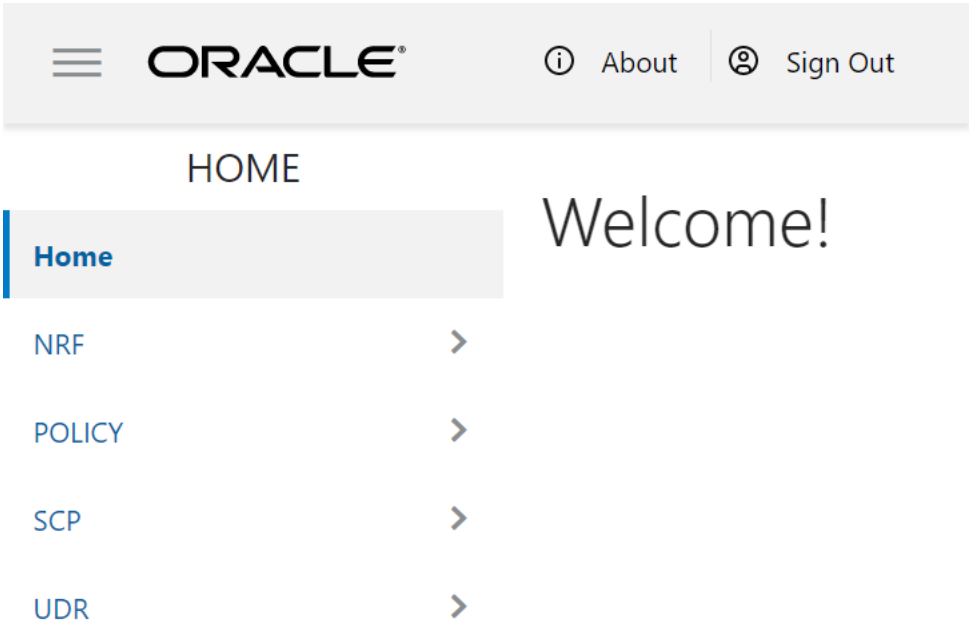
CNC Console Interface

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 5-1 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.

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.

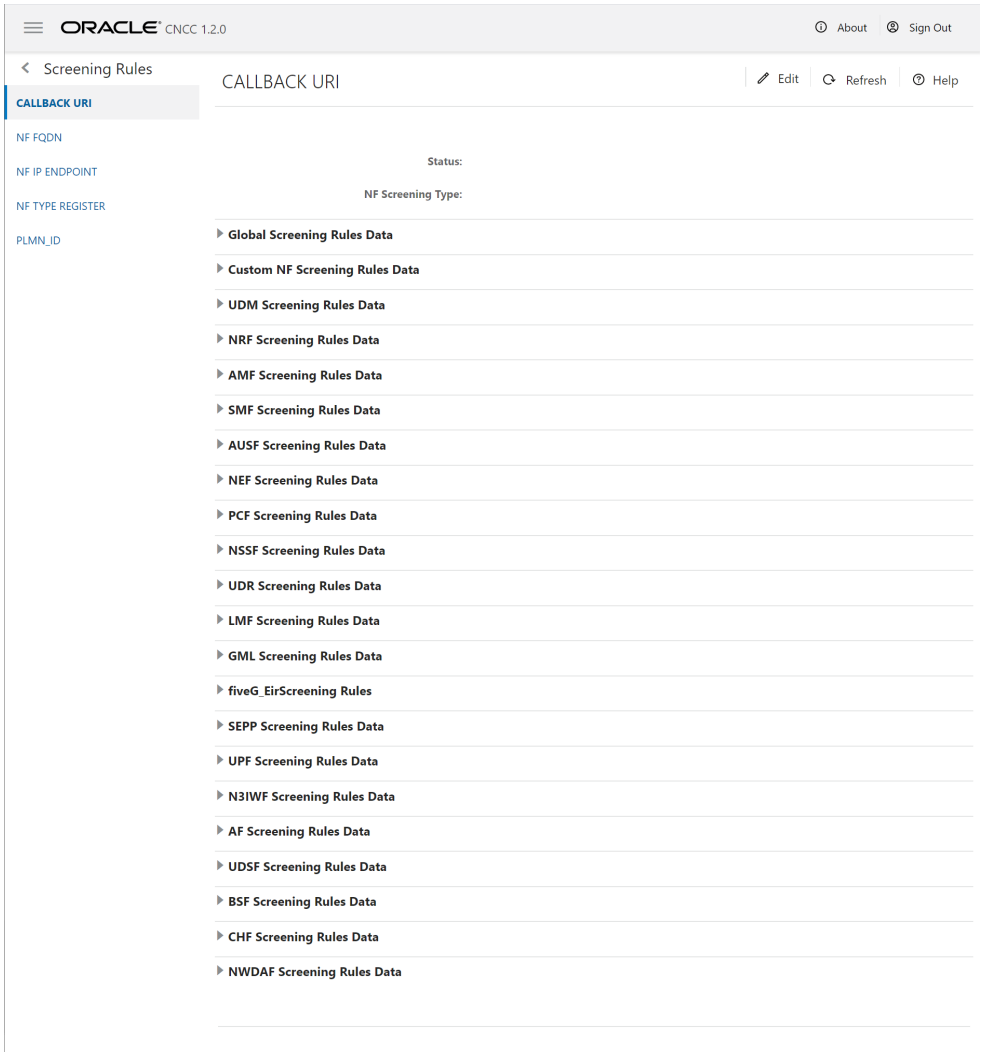
NRF screens the callback URI present in the request before allowing access to management service. Host present in callback URI (FQDN+port or IP+port) must be used for screening. In CALLBACK URI, the attributes that can be modified are FQDN, Port and IP address.

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 5-2 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.

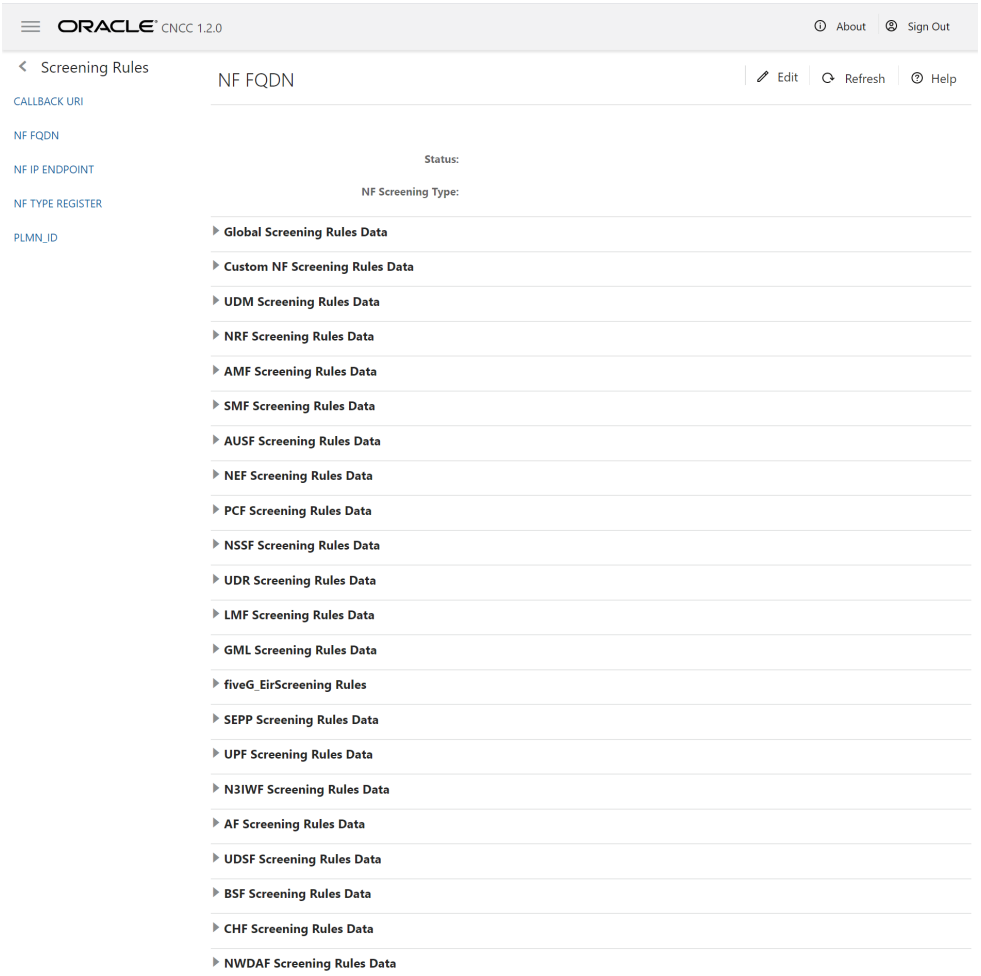
In NF FQDN, the attributes that can be modified are pattern, fqdn in NFProfile and fqdn in NFService.

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 5-3 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*. Refer to NF FQDN Parameters for more information in parameter values and description.
4. Click **Save**.

Refer to Configuring NF Screening for more information on parameter values and description.

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

NF IP Endpoint

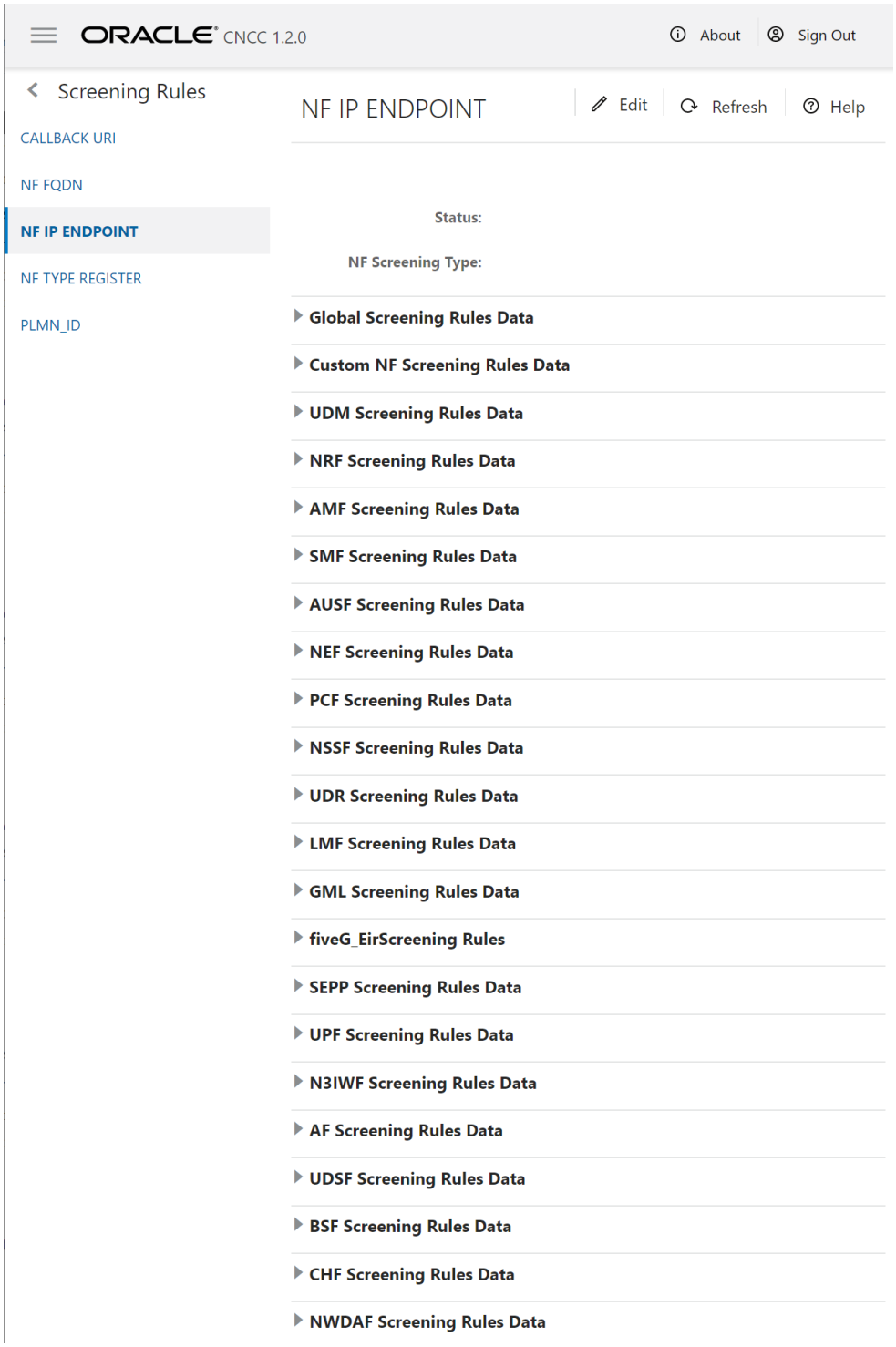
NRF screens the IP endpoint(s) present in the request before allowing access to management service.

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 5-4 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

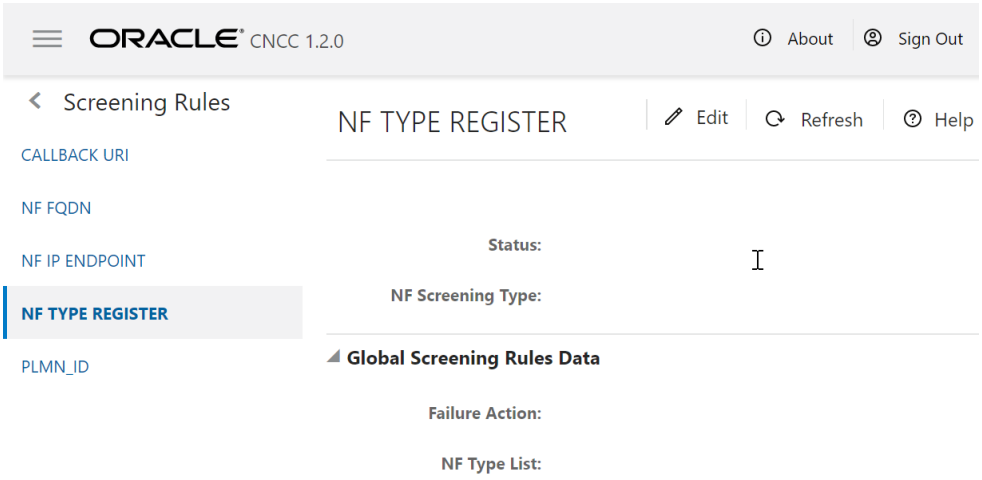
NRF screens the NF type present in the in-coming service request.

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 5-5 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

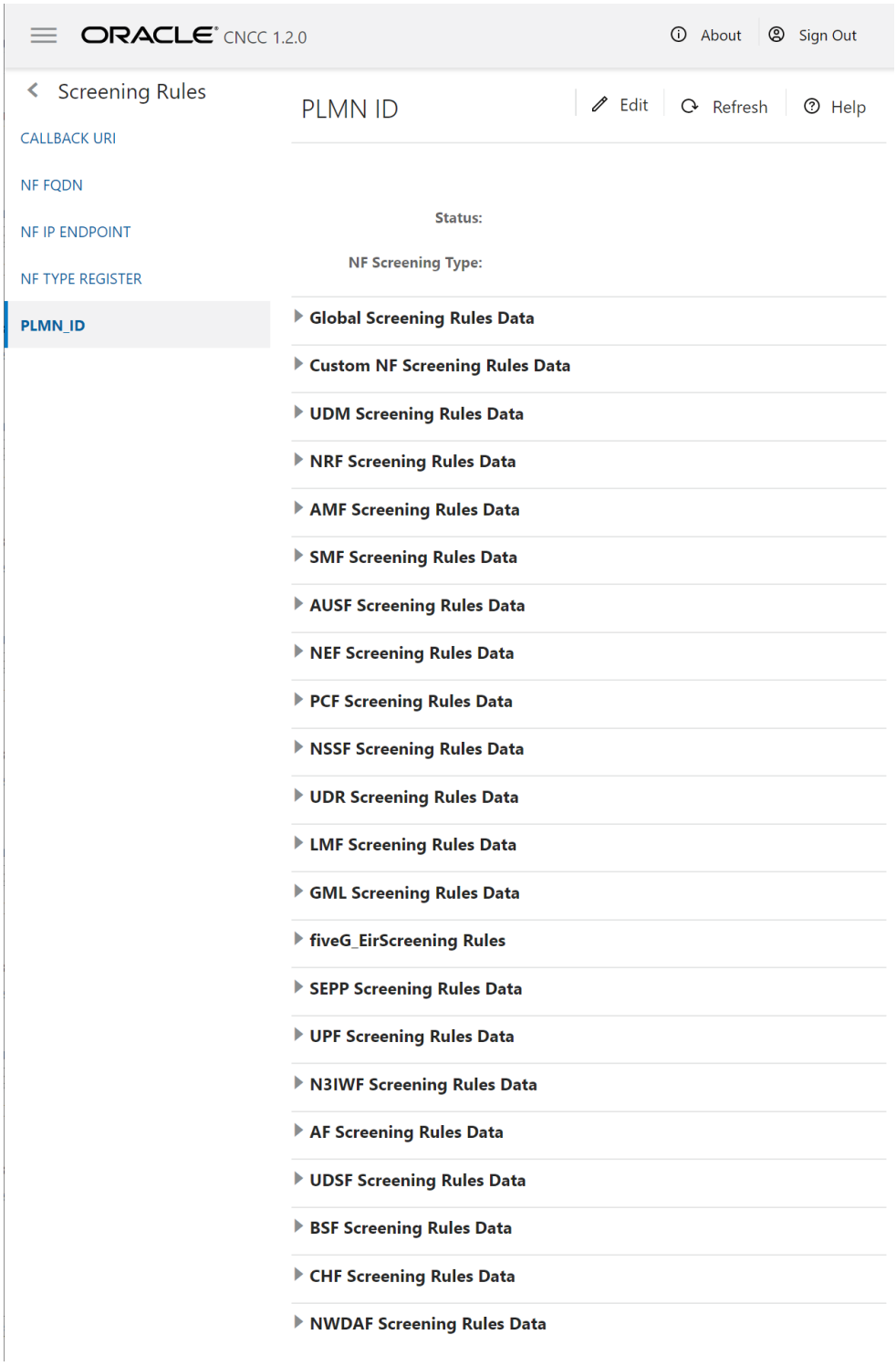
NRF screens the PLMN Id present in the request before allowing access to management service.

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 5-6 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.

System Options

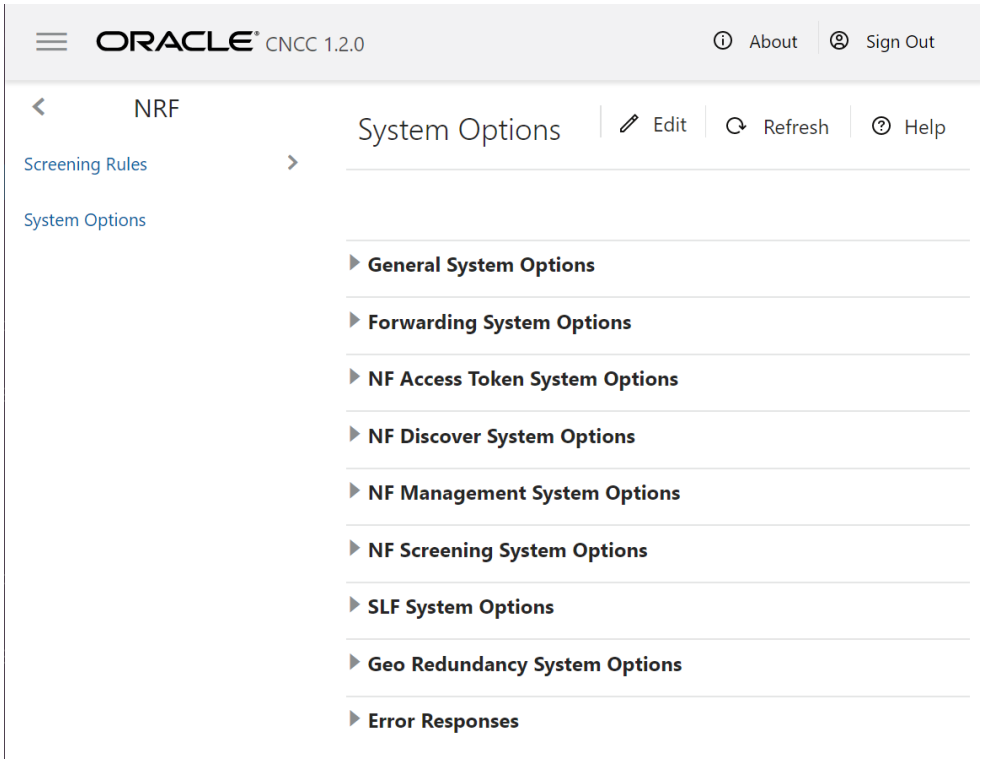
This section explains the procedure to configure system options.

Configuring System Options parameters

To configure system options parameters follow the procedure:

1. From the left navigation menu, navigate to **NRF** and then **Screening Rules**, select **System Options**.

Figure 5-7 System Options



2. Click **Edit** from the top right side to edit or update a system options parameters. The screen is enabled for modification.
3. Enter the values for the attributes as per the requirement. Refer to General Configurations for more information in parameter values and description.
4. Click **Save**.

Modifying Configuration list

The user can add, edit or delete the Configuration list such as NRF PLMN , Forwarding System Option, SLF Host Config, SLF Error Responses parameters or NRF Forwarding Error Responses.

Adding Configuration list

To add a Configuration list:

1. Click **Edit** from the top left corner of the System Options screen. The **Edit System Options** Screen is enabled to edit.
2. Click **Add** from the top left of the Configuration list table for each *System Options* section.
3. Enter the attribute values. Refer to General Configurations for more information in parameter values and description.
4. Click **Save**.

Editing Configuration list

To edit an existing configuration list:

1. Click **Edit** from the top left corner of the System Options screen. The **System Options** Screen is enabled to edit.
2. Click **Edit** from the Configuration list. Refer to General Configurations for more information in parameter values and description.
3. Modify the attribute values.
4. Click **Save**.

Deleting Configuration list

To delete a Configuration list:

1. Click **Edit** from the top left corner of the System Options screen. The **System Options** Screen is enabled to edit.
2. Click **Delete** from the action items.
   The *" Do you want to delete the record"* message appears.
3. Click **OK** to delete the parameter.
4. Click **Save**.

6
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 6-1 Dimensions Legend

Dimension	Details
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 (e.g. "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 6-1 (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.

Table 6-2 OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
1	Total number of ingress requests	Total number of requests received at OCNRF	oc_ingressgateway_http_requests_total		

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
2	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"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running
3	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"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
4	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"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running
5	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"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
6	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*."}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the Kubernetes worker node on which micro-service is running
7	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"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the Kubernetes worker node on which micro-service is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
8	NF Subscribe Success	Total number of successful NFSubscribe service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="201 CREATED",Route_path=~".*nnrf-nfm/v1/subscriptions.*", Method="POST"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the Kubernetes worker node on which micro-service is running
9	NF Unsubscribe Success	Total number of successful NFUnSubscribe service operations at OCNRF	oc_ingressgateway_http_responses_total{Status="204 NO_CONTENT", Route_path=~".*nnrf-nfm/v1/subscriptions.*", Method="DELETE"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the Kubernetes worker node on which micro-service is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
10	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"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the Kubernetes worker node on which micro-service is running
11	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.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
12	4xx Responses (Subscriptions)	Total number of 4xx responses(NfSubscribe/ NfUnsubscribe)	oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-nfm/v1/subscriptions.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running
13	4xx Responses (Discovery)	Total number of 4xx responses(NfDiscover)	oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*nnrf-disc/v1/nf-instances.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
14	4xx Responses (AccessToken)	Total number of 4xx responses(NfAccessToken)	oc_ingressgateway_http_responses_total{Status=~"4.*",Route_path=~".*oauth2/token.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running
15	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.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
16	5xx Responses (Subscriptions)	Total number of 5xx responses(NfSubscribe/ NfUnsubscribe)	oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-nfm/v1/subscriptions.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running
17	5xx Responses (Discovery)	Total number of 5xx responses(NfDiscover)	oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*nnrf-disc/v1/nf-instances.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
18	5xx Responses (AccessToken)	Total number of 5xx responses(NfAccessToken)	oc_ingressgateway_http_responses_total{Status=~"5.*",Route_path=~".*oauth2/token.*"}		Method- HTTP method of request Status - status code in HTTP response Uri- URI from the request line Node-Name of the kubernetes worker node on which microservice is running
19	NfRegistrations Total	Number of Registration Requests received	ocnrf_nfRegister_rx_requests_total	NfRegistrations Total	NrfLevel NfInstanceId RequesterNfType
20	NfRegistrations Responses Total	Number of Registration Responses sent.	ocnrf_nfRegister_tx_responses_total	NfRegistrations Responses Total	NrfLevel NfInstanceId RequesterNfType HttpStatusCode
21	NfRegistrations Per Service Total	Number of Registrations received and processed successfully per Service.	ocnrf_nfRegister_rx_requests_success_perService_total	NfRegistrations Per Service [serviceName :- {{ serviceName }}, nfInstanceId :- {{NfInstanceId}}]	NrfLevel NfInstanceId ServiceName ServiceInstanceId

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
22	NFUpdates Total	Number of Update Requests received.	ocnrf_nfUpdate_rx_requests_total	NfUpdates Total	NrfLevel NfInstanceId RequesterNfType UpdateType(Partial/Complete)
23	NFUpdates Responses Total	Number of Update Responses sent.	ocnrf_nfUpdate_tx_responses_total	NfUpdates Responses Total	NrfLevel NfInstanceId RequesterNfType UpdateType(Partial/Complete) HttpStatusCode
24	NFUpdates Per Service Total	Number of NfUpdates received and processed successfully per Service.	ocnrf_nfUpdate_rx_requests_success_perService_total	NFUpdates Per Service [serviceName :- {{ serviceName }}, serviceInstanceId:- {{ServiceInstanceId}}]	NrfLevel, Updatetype =(Partial/Complete), NfInstanceId , ServiceName, ServiceInstanceId
25	Heartbeat Requests Total	Number of Heartbeat Requests received	ocnrf_nfHeartbeat_rx_requests_total		NrfLevel NfInstanceId RequesterNfType
26	Heartbeat Resposnes Total	Number of Heartbeat Responses sent	ocnrf_nfHeartbeat_tx_responses_total		Nrflevel, NfInstanceId , RequesterNfType , HttpStatusCode
27	NF De-Registration Requests Total	Number of De-registration requests received	ocnrf_nfDeregister_rx_requests_total		NrfLevel, NfInstanceId , RequesterNfType

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
28	NF De-Registration Responses Total	Number of De-registration responses sent	ocnrf_nfDeregister_tx_responses_total		NrfLevel, NfInstanceId, RequesterNfType , HttpStatusCode
29	NF De-Registrations Per Service Total	Number of De-registration requests received and process successfully per Service	ocnrf_nfDeregister_rx_requests_success_perService_total	NFDeregistration Per Service [serviceName :- {{ serviceName }}, serviceInstanceId:- {{ServiceInstanceId}}]	NrfLevel, ServiceName, ServiceInstanceId, NfInstanceId
30	NF List Retrieval Requests Total	Number of NFListRetrieval requests received	ocnrf_nfListRetrieval_rx_requests_total		NrfLevel, RequesterNfType
31	NF List Retrieval Responses Total	Number of NFListRetrieval responses sent	ocnrf_nfListRetrieval_tx_responses_total		NrfLevel, RequesterNfType , HttpStatusCode
32	NF Profile Retrieval Requests Total	Number of NFProfileRetrieval requests received	ocnrf_nfProfileRetrieval_rx_requests_total		NrfLevel, NfInstanceId
33	NF Profile Retrieval Responses Total	Number of NFProfileRetrieval responses sent	ocnrf_nfProfileRetrieval_tx_responses_total		NrfLevel, NfInstanceId, HttpStatusCode
34	Number of Heartbeats missed	Number of heartbeats missed.	ocnrf_heartbeat_missed_total		NrfLevel, RequesterNfType , NfInstanceId

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
35	NF Status Subscribe Requests Total	Number of NStatusSubscribe requests received	ocnrf_nfStatusSubscribe_rx_requests_total		NrfLevel, RequesterNfType, OperationType
36	NF Status Subscribe Responses Total	Number of NfStatusSubscribe responses sent	ocnrf_nfStatusSubscribe_tx_responses_total		NrfLevel, RequesterNfType , HttpStatusCode, OperationType
37	NF Status UnSubscribe Requests Total	Number of NfStatusUnsubscribe requests received	ocnrf_nfStatusUnsubscribe_rx_requests_total		NrfLevel, RequesterNfType
38	NF Status UnSubscribe Responses Total	Number of NfStatusUnsubscribe responses sent	ocnrf_nfStatusUnsubscribe_tx_responses_total		NrfLevel, RequesterNfType, HttpStatusCode
39	NF Status Notifications Requests Sent	Number of NfStatusNotify requests sent	ocnrf_nfStatusNotify_tx_requests_total		NrfLevel, NotificationEventType, TargetNfType
40	NF Status Notifications Responses Received	Number of NfStatusNotify responses received	ocnrf_nfStatusNotify_rx_responses_total		NrfLevel, NotificationEventType, TargetNfType, HttpStatusCode
41	NF Status Notifications Requests Failed	Number of NfStatusNotify requests failed to sent out	ocnrf_nfStatusNotify_requests_failed_total		NrfLevel, NotificationEventType, TargetNfType

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
42	NfDiscover Requests Total	Number of NfDiscover Requests received	ocnrf_nfDiscover_rx_requests_total	NfDiscover Req [TargetNf :- {{ TargetNfType }}, RequesterNfType :- {{RequesterNfType}}]	NrfLevel, TargetNfType, RequesterNfType
43	NfDiscover Responses Total	Number of NfDiscover responses sent	ocnrf_nfDiscover_tx_responses_total		NrfLevel, TargetNfType, RequesterNfType, HttpResponseCode
44	NFDiscover Per Service Total	Number of NfDiscover requests received and processed successfully per Service	ocnrf_nfDiscover_rx_requests_success_perService_total	NFDiscover Per Service [serviceName :- {{ serviceName }}]	NrfLevel, RequesterNfType, ServiceName
45	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	Discovered profiles [TargetNfType :- {{TargetNfType}}, Bucket :- {{ Bucket }}]	NrfLevel, TargetNfType, BucketSize NfFqdn
46	Active Registrations	Number of active registered NFs at any point of time	ocnrf_active_registrations_count	Active Registrations [NfType- {{ NfType }}, NrfLevel- {{ NrfLevel }}]	NfType, NrfLevel

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions
47	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/ NfSubscribe/ NfUnsubscribe/ **NfAccessToken**) **Note**: Latency calculated by this metric doesn't include time taken by OCNRF API gateway.	ocnrf_message_processing_time_seconds	Avg NRF Latency {{ ServiceOperation }} {{ RequesterNfType }}	NrfLevel,RequesterNfType ,ServiceOperation
48	OCNRF database operations	Database operation count corresponding to every service operation		ocnrf_dbmetric_total	Method, DBOperation, NrfLevel, HttpStatusCode
49	Database operation round trip time	Time (in microseconds) taken by database operation corresponding to every service operation NfRegister/ NfUpdate/ NfDelete/ NfProfileRetrieval /NfListRetrieval/ NfHeartbeat/ NfDiscover/ NfSubscribe/ NfUnsubscribe/ NfAccessToken)	ocnrf_dbmetrics_round_trip_time_seconds		• Method • DBOperation • ServiceOperation • TableName: (NRF Table Names) • NrfLevel • HttpStatusCode

In the above NRF Metrics table, 4xx and 5xx are the error codes in REST API.

Table 6-3 NF Screening specific metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions	Notes
1	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	NFRegister, NFUpdate	NRF level NF type	See Note 1 below this table.
2	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	NFRegister, NFUpdate	NRF level NF type	See Note 1 below this table.
3	Total NF Requests for which Screening Failed	The total number of requests for which screening failed against NF IP endpointscreening list.	ocnrf_nfScreening_nfIpEndPoint_requestFailed_total	NFRegister, NFUpdate	NRF level NF type	See Note 1 below this table.
4	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	NFRegister, NFUpdate	NRF level NF type	See Note 1 below this table.
5	Total NF Requests for which Screening Failed	The total number of requests for which screening failed against Callback URIscreening list.	ocnrf_nfScreening_callbackUri_requestFailed_total	NFRegister, NFUpdate, NFSubscribe	NRF level NF type	See Note 1 below this table.
6	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	NFRegister, NFUpdate, NFSubscribe	NRF level NF type	See Note 1 below this table.
7	Total NF Requests for which Screening Failed	The total number of requests for which screening failed against PLMN idscreening list.	ocnrf_nfScreening_plmnId_requestFailed_total	NFRegister, NFUpdate	NRF level NF type	See Note 1 below this table.

Table 6-3 (Cont.) NF Screening specific metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions	Notes
8	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	NFRegister, NFUpdate	NRF level NF type	See Note 1 below this table.
9	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	NFRegister	NRF level NF type	See Note 1 below this table.
10	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	NFRegister	NRF level NF type	See Note 1 below this table.
11	NF Screening not applied Internal Error	The total number of times screening not applied due to internal error.	ocnrf_nfScreening_notApplied_InternalError_total	NFRegister, NFUpdate, NFSubscribe	NRF level NF type	See Note 1 below this table.

Note:

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

NF Access token metrics

Table 6-4 NF Access token metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
1	NF Access Token Request Received Total	The total number of access token requests received	ocnrf_accessToken_rx_requests_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel

Table 6-4 (Cont.) NF Access token metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
2	NF Access Token Responses Sent Total	The total number of access token responses sent	ocnrf_accessToken_tx_responses_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode
3	NF Access Token Request Rejected (ClientNotAuthorized)	Number of access token request for which client authorized failed RejectionReason = ClientNotAuthorized	ocnrf_accessToken_tx_rejected_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, RejectionReason HttpStatusCode RejectionReason = ClientNotAuthorized
4	NF Access Token Request Rejected (ProducerWithRequestedScopeNotFound)	Number of access token not granted because of no producer instance registered for service/s in the scope RejectionReason = ProducerWithRequestedScopeNotFound	ocnrf_accessToken_tx_rejected_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, RejectionReason HttpStatusCode RejectionReason = ProducerWithRequestedScopeNotFound
5	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. RejectionReason = ProducerWithRequestedNfInstanceIdNotFound	ocnrf_accessToken_tx_rejected_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, RejectionReason HttpStatusCode RejectionReason = ProducerWithRequestedNfInstanceIdNotFound

Table 6-4 (Cont.) NF Access token metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
6	NF Access Token Request Rejected (InconsistentScope)	Number of access token not granted because services in the scope belong to different NF types. RejectionReason = InconsistentScope	ocnrf_accessToken_tx_rejected_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, RejectionReason HttpStatusCode RejectionReason = InconsistentScope
7	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. RejectionReason = ConsumerNFTypeMismatch	ocnrf_accessToken_tx_rejected_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, RejectionReason HttpStatusCode RejectionReason = ConsumerNFTypeMismatch
8	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. RejectionReason = ProducerNFTypeMismatch	ocnrf_accessToken_tx_rejected_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, RejectionReason HttpStatusCode RejectionReason = ProducerNFTypeMismatch
9	NF Access Token Request Rejected (InternalError)	Number of access token not granted because failure at NRF due to internal error. RejectionReason = InternalError	ocnrf_accessToken_tx_rejected_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode RejectionReason = ProducerNFTypeMismatch

Table 6-4 (Cont.) NF Access token metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
10	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	Access Token	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode RejectionReason = ConsumerNfTypeNotAllowed
11	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	Access Token	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel, HttpStatusCode RejectionReason = ConsumerPlmnNotAllowed

NRF-SLF specific metrics

Table 6-5 NRF-SLF specific metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
1	Discover Request Received For SLF Total	The total number of NF Discover request received for SLF	ocnrf_nfDiscover_ForSLF_rx_requests_total	NFDiscover	TargetNfType, NRFLevel

Table 6-5 (Cont.) NRF-SLF specific metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
2	Discover Response Sent For SLF Total	The total number of NF Discover responses sent for SLF	ocnrf_nfDiscover_ForSLF_tx_responses_total	NFDiscover	TargetNfType, NRFLevel, HttpStatusCode, RejectionReason Possible Reject reasons:- RejectionReason = SLFCommunicationFailure RejectionReason = MandatoryParamsMissing RejectionReason = SLFConfigurationMissing RejectionReason = GroupIdNotFound RejectionReason = ErrorFromSLF RejectionReason = InternalError RejectionReason = *NotApplicable *NotApplicable is applicable for 2xx Status code
3	SLF Query Requests Sent Total	The total number of SLF query request sent	ocnrf_SLF_tx_requests_total	NFDiscover	TargetNfType, NRFLevel, SubscriptionIdType
4	SLF Query Responses Received Total	The total number of SLF query response received	ocnrf_SLF_rx_responses_total	NFDiscover	TargetNfType, NRFLevel, SubscriptionIdType,HttpStatusCode, GroupId
5	SLF Round Trip Time Total	Time (in microseconds) after sending query to SLF and getting response from SLF	ocnrf_slf_round_trip_time_seconds	NFDiscover	TargetNfType, SubscriptionIdType, HttpStatusCode, GroupId, NrfLevel, SLF ApiRoot

NRF Forwarding Metrics

Table 6-6 NRF Forwarding Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
1	NF Access Token Requests Forwarded Total	The total number of Access Token Request forwarded to Primary/ Secondary NRF	ocnrf_forward_accessToken_tx_requests_total	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel
2	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	Access Token	TargetNfType, ClientNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel,HttpStatusCode, RejectionReason **RejectionReason**: • InternalError • NRFCommunicationFailure • ErrorFromNRF • NRFForwardingConfigurationMissing • LoopDetected *NotApplicable is applicable for 2xx Status code
3	NF Profile Retrieval Requests Forwarded Total	The total number of Profile Retrieval Request forwarded to Primary/ Secondary NRF	ocnrf_forward_nfProfileRetrieval_tx_requests_total	NFProfileRetrieval	NrfLevel, NfInstanceId

Table 6-6 (Cont.) NRF Forwarding Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
4	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	NFProfileRetrieval	NrfLevel, NfInstanceId, HttpStatusCode, RejectionReason **RejectionReason**: • InternalError • NRFCommunicationFailure • ErrorFromNRF • NRFForwardingConfigurationMissing • LoopDetected *NotApplicable is applicable for 2xx Status code
5	NF Status Subscribe Forwarded Requests Total	The total number of Status Subscribe Request forwarded to Primary/ Secondary NRF	ocnrf_forward_nfStatusSubscribe_tx_requests_total	NFStatusSubscribe, NFStatusUnsubscribe	NrfLevel, RequesterNfType, OperationType
6	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	NFStatusSubscribe, NFStatusUnsubscribe,	NrfLevel, RequesterNfType, HttpStatusCode, OperationType, RejectionReason **RejectionReason**: • InternalError • NRFCommunicationFailure • ErrorFromNRF • NRFForwardingConfigurationMissing • LoopDetected *NotApplicable is applicable for 2xx Status code

Table 6-6 (Cont.) NRF Forwarding Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
7	NF Discovery Forwarded Requests Total	The total number of NF Discovery Request forwarded to Primary/ Secondary NRF	ocnrf_forward_nfDiscover_tx_requests_total	NFDiscover	NrfLevel, TargetNfType, RequesterNfType
8	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	NFDiscover	NrfLevel, TargetNfType, RequesterNfType, HttpResponseCode, RejectionReason **RejectionReason**: • InternalError • NrfCommunicationFailure • NrfForwardingConfigurationMissing • LoopDetected ErrorFromNrf *NotApplicable is applicable for 2xx Status code
9	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	NFStatusSubscribe, NFStatusUnsubscribe, NFProfileRetrieval, NFDiscover, Access Token	NrfLevel, RequesterNfType, ServiceOperation

GeoRedundancy metrics

Table 6-7 GeoRedundancy metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
1.	DB Replication status	The current replication status of the db tier service.	ocnrf_dbreplication_status	NA	NrfLevel,DbReplicationStatus
2.	DB Replication down Time	Time taken for the replication status to change from "INACTIVE" to "ACTIVE"	ocnrf_dbreplication_down_time_seconds	NA	NrfLevel,DbReplicationDownStartTime,DbReplicationDownEndTime
3.	Total NfInstances switched over from mated site	The number of NFInstances that got switched over from the mated site.	ocnrf_nf_switch_over_total	NfRegister, NfUpdate,NfDeregister, NfHeartbeat	NrfLevel, NfInstanceId,RemoteNrfInstanceId,ServiceOperation,OperationType
4.	Total NfSubscriptions switched over from mated site	The number of NfSubscriptions that got switched over from the mated site.	ocnrf_nfSubscriptions_switch_over_total	NfStatusSubscribe,NfStatusUnsubscribe, NrfAuditor	NrfLevel,SubscriptionId,RemoteNrfInstanceId,ServiceOperation,OperationType
5.	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	NA	NrfLevel, NfInstanceId, NfStatus
6.	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	NA	NrfLevel,NfSubscriptionId,SubscriptionStatus
7.	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	NA	NrfLevel, NfInstanceId, NfStatus, HeartbeatTimer

Table 6-7 (Cont.) GeoRedundancy metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
8	Total NfSubscriptions whose validityTime has expired	The number of NfSubscriptions whose validityTime has expired	ocnrf_nfSubscriptions_expired_total	NA	NrfLevel,SubscriptionId

NF AccessToken Authorization Metrics

Table 6-8 NF AccessToken Authorization Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
1	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	NfAccessToken	TargetNfType, RequesterNfType, TargetNfInstanceId, ClientNfInstanceId, Scope, NrfLevel,HttpStatusCode RejectionReason = ClientNotAuthorized

NF Authentication Metrics

Table 6-9 NF Authentication Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
1	NF Authentication Failure Total	The total number of request for which FQDN based Authentication failed at OCNRF	ocnrf_nf_authentication_failure_total	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 6-10 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

Table 6-10 (Cont.) KPI Details

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response code
NF Register Success		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
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

Table 6-10 (Cont.) KPI Details

KPI Name	KPI Details	Metric used for KPI	Service Operation	Response code
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 6-11 Alert Details

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

Table 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }} : NFDiscovery service unavailable.'	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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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**: '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 6-11 (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.kubernetes_pod_name}}, 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 6-11 (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.kubernetes_pod_name}}, 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 6-11 (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 6-11 (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: {{ $value }})' **summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: Traffic Rate is above 80 Percent of Max requests	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 6-11 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			per second(1000)'				

Table 6-11 (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: {{ $value }})' **summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: Traffic Rate is above 90 Percent of Max requests per	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 6-11 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			second(1000)'				

Table 6-11 (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: {{ $value }})' **summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }} {{ . | first | value | humanizeTimestamp }} {{ end }}: Traffic Rate is above 95 Percent of Max requests per	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 6-11 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
			second(1000)'				

Table 6-11 (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 {{ $value }})' **summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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 {{ $value }})'**summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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 {{ $value }})' **summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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 {{ $value }})' **summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, 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 6-11 (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 {{ $value }})' **summary**: 'namespace: {{$labels.kubernetes_namespace}}, nftype: {{$labels.NfType}}, nrflevel: {{$labels.NrfLevel}}, podname: {{$labels.kubernetes_pod_name}}, 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.	

Table 6-11 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OCNRF Application Alerts							

Table 6-11 (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: {{ $value }})' **summary**: 'namespace: {{$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 critical	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 6-11 (Cont.) Alert Details

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

Table 6-11 (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: {{ $value }})' **summary**: 'namespace: {{$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 6-11 (Cont.) Alert Details

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

Table 6-11 (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: {{ $value }})' **summary**: 'namespace: {{$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 6-11 (Cont.) Alert Details

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

Table 6-11 (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: {{ $value }})' **summary:** 'namespace: {{$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 6-11 (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.	Critical	description: 'The Database Replication Status is currently INACTIVE.' summary: '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 6-11 (Cont.) Alert Details

Alert	Trigger Condition	Severity	Alert details provided	OID	Metric Used	Resolution	Notes
OcnrfAccessTokenRequestsRejected	OCNRF rejected an AccessToken Request	critical warning	**description**: 'AccessToken request(s) have been rejected by OCNRF (current value is: {{ $value }})' **summary**: 'namespace: {{$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 6-11 (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	critical warning	**description**: 'Service request(s) received from NF have been rejected by OCNRF (current value is: {{ $value }})' **summary**: 'namespace: {{$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 Rejection Reason shall be present in the alert	

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 OHC. 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 6-12 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
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

Table 6-12 (Cont.) Alerts

Alert Name	Details	Default Value	Notes
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. Please use the NrfAlertRules.yaml file updated in OCNRF Alert configuration section.

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-1.8.0.mib
  This is considered as OCNRF top level mib file, where the Objects and their data types are defined.
- OCNRF-MIB-1.8.0.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 OHC. Refer to OCNRF Installation and Upgrade guide for more details.