Oracle® Communications

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

Release 1.7.2

F32542-02

August 2020

Oracle Communications Network Repository Function (NRF) Cloud Native User's Guide, Release 1.7.2

F32542-02

Contents

List of Figures

List of Tables

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

New and Updated Features in Release 1.7.2

- Added OCNRF Access Token Service Usage Details chapter.

New and Updated Features in Release 1.7.0

- Added **geoRedundancySystemOptions** parameters in General Configurations.
- Updated time units, range and values for parameters in General Configurations.
- OCNRF's host configuration is moved from helm based to rest based configuration.
- Added GeoRedundancy feature related metrics and alerts in OCNRF Metrics and OCNRF Alerts respectively.
- Updated metrics dimensions in OCNRF Metrics.
- Added allowDuplicateSubscriptions parameter in generalConfiguration. By default, OCNRF will allow duplicate subscriptions. Refer to General Configurations table for more information.
- Updated minimum supported heartbeat value (nfHeartBeatTimer) value. Refer to General Configurations table for more info.

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.

The OCNRF is a key component of the 5G Service Based Architecture. The 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, the OCNRF also supports discovery mechanisms that allows NFs to discover each other and get updated status of the desired NFs.

The 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 1.5 Installation Guide
- OCNRF Installation and Upgrade Guide
- CNC Console User's Guide
- ATS User Manual

2

OCNRF Supported Services

This section includes information about the services supported by OCNRF.

OCNRF supports the following services:

OCNRF Management Services

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

OCNRF supports the following management services:

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

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

OCNRF Discovery Service

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

OCNRF supports the following discovery service:

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

OCNRF Access Token Service

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

OCNRF supports the following access token service:

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

3 OCNRF Architecture

OCNRF comprises of various microservices deployed in Kubernetes based Cloud Native Environment (CNE, example: OCCNE). Some common services like logs or metrics data collection, analysis and graphs or charts visualization, and so on are provided by the environment. The microservices integrate with them and provide them 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 (i.e. NF Registration microservice, NF Subscription microservice, NRF Auditor microservice). In case any of them are down, status of NF Management combined together will be down.

  This microservice is also responsible to fetch DB replication status whether it is active or not-active. This happens only if `geoRedundancyFeatureStatus` is ENABLED.

OCNRF Features

Following are 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 (SLF)

OCNRF supports 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

This 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 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.

4
Configuring OCNRF

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 Configuration

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.

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 configuring 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
},
"nfScreeningSystemOptions": {
    "nfScreeningFeatureStatus": "DISABLED",
    "nfScreeningFailureHttpCode": 403
},
"nfAccessTokenSystemOptions": {
    "oauthTokenAlgorithm": "ES256",
    "oauthTokenExpiryTime": "1h",
    "authorizeRequesterNf": "ENABLED",
    "logicalOperatorForScope": "AND",
    "audienceType": "NF_INSTANCE_ID"
},
"nfManagementSystemOptions": {
    "nfHeartBeatTimer": "1m30s",
    "nfHeartBeatMissAllowed": 3,
    "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"
},
"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"
 }

}
```

Data Model

Note:

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

Presence in the JSON BODY in PUT HTTP method means any attribute(s) can be updated individually or together.

O - Optional

M - Mandatory

C - Conditional

Table 4-5 NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
generalSystemOptions	nrfPlmnList	array (PlmnId)		O		This value shall have at least one PLMN supported by OCNRF and this value shall be set before using OCNRF. See the footnote.
generalSystemOptions	enableF3	ENUM (true or false)	true or false	O	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	O	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	O	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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
generalSystemOptions	defaultPriority	INTEGER	0 - 65535	O	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	O	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	O	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.
generalSystemOptions	maximumHopCount	INTEGER	1-5	O	3	Maximum number of Nodes (SLF/NRF's) that OCNRF can communicate, to service a request.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
generalSystemOptions	ocnrfEndPointHost	STRING	None	O	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	O	80	OCNRF EndPoint Host's Port

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
forwardingSystemOptions	nrfHostConfig	array (NFConfig)		O		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	O	"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	M/O/C	Default Values	Description
forwardingSystemOptions	profileRetreivalForwardingStatus	String (Feature Status)		O	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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
forwardingSystemOptions	subscriptionForwardingStatus	String (Feature Status)		O	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	M/O/C	Default Values	Description
forwardingSystemOptions	discoveryForwardingStatus	String (Feature Status)		O	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)		O	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)		O	DISABLED	This attribute indicates if NF Screening Feature is enabled or not. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
nfScreeningSystemOptions	nfScreeningFailureHttpCode	INTEGER		O	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.
nfManagementSystemOptions	nfHeartBeatTimer	String	10s - 5m	O	1m30s	If Heartbeat timer value is not received in NFProfile during NFRegister, this default value will be used by OCNRF. If Heartbeat timer value is received in NFProfile during NFRegister, minimum value will be used for validation and limit purpose. It means if value provided less than minimum value, then minimum value will be taken as heartbeat timer value. If Heartbeat timer value is received in NFProfile during NFRegister, maximum value of range will be used for validation and limit purpose. It means if value provided more than maximum value, then maximum value will be taken as heartbeat timer value.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	nfHeartBeatMissAllowed	INTEGER	0 - 15	O	3	Indicates the allowed number of HeartBeat miss after which the NFProfile is marked as suspended. If the value is set to 0, NF profiles for which even single heartbeat is missed will be marked as suspended. See the footnote.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
nfManagementSystemOptions	nfNotifyLoadThreshold	INTEGER	0 - 99	O	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	O	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.
nfManagementSystemOptions	subscriptionValidityDuration	String	10s - 720h	O	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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
nfManagementSystemOptions	nrfSupportForProfileChangesInNotification	ENUM (true, false)	true or false	O	false	OCNRF sends profileChanges attribute instead of NFProfile in Notification, if this flag is enabled. See the footnote.
nfManagementSystemOptions	nfProfileSuspendDuration	String	10s - 744h	O	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	O	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.
nfManagementSystemOptions	allowDuplicateSubscriptions	ENUM (true, false)	true or false	O	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	O	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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
nfDiscoverSystemOptions	profilesCountInDiscoveryResponse	INTEGER	0 - 20	C	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	C	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)		O	ES256	Access token key algorithm which will be used to sign the oauth token. See the footnote.
nfAccessTokenSystemOptions	oauthTokenExpiryTime	String	1s - 168h	O	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.
nfAccessTokenSystemOptions	authorizeRequesterNf	String (Feature Status)		O	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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
nfAccessTokenSystemOptions	audienceType	String (AudienceType)		O	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)		O	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.
slfSystemOptions	slfFeatureStatus	String (FeatureStatus)		O	DISABLED	Enables/disables the SLF Feature. See NOTE 1.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
slfSystemOptions	slfHostConfig	array (NfConfig)		C		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.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
slfSystemOptions	supportedNfTypeList	array		C		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 shall also be set. See the footnote.
slfSystemOptions	preferredSubscriberIdType	String (SubscriberIdType)	SUPI or GPSI	O	SUPI	This attribute will only be used, in case different type of subscriber identifiers (SUPI, GPSI) are present in NFDiscover service operation message, which subscriber identifier shall be used for the query to SLF. See the footnote.
slfSystemOptions	rerouteOnResponseHttpStatusCodes	ResponseHttpStatusCodes		O	"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)		O	DISABLED	Enables/Disables the geoRedundancy feature in OCNRF. See the footnote.
geoRedundancySystemOptions	replicationLatency	String	1s - 10m	O	5s	This attribute defines the time taken for the data in the database to get replicated between GeoRedundant OCNRFs. The value is in pHqMrS format. Where p,q,r are integers and H,M,S or h,m,s denote hours, minutes & seconds respectively.

Table 4-5 (Cont.) NrfSystemOptions - Parameters

Parent Attribute Name	Attribute Name	Data Type	Constraints	M/O/C	Default Values	Description
geoRedundancySystemOptions	monitorNrfServiceStatusInterval	String	1s - 10s	O	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	O	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.
errorResponses	slfErrorResponses	array (ErrorInfo)		O		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)		O		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.

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

Table 4-11 (Cont.) NFConfig

Attribute	DataType	Presence	Description
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
error_condition	ErrorCondition	ReadOnly	Error Conditions
error_response_code	Integer	M	This response code will be used when corresponding error condition will occur.
error_response_description	String	M	This response description will be used when corresponding error condition will occur.

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

Table 4-15 ResponseHttpStatusCodes

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

Configuring NF Screening

This section provides information for configuring NF Screening.

Table 4-16 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 4-17 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 4-18 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 4-19 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 4-20 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 4-21 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 4-22 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 4-23 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 4-24 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 4-25   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 4-25 (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 4-25 (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 4-26 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 4-27 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 4-28 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 4-29 NfScreeningRulesListStatus - Parameters

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

Table 4-30 FailureAction - Parameters

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

Table 4-31 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 4-32 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 4-32 (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 4-33 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.
ipv4Address Range	Ipv4Address Range	C	Range of IPv4 addresses.
ipv6Address	Ipv6Addr	C	IPv6 address to be matched.
ipv6Address Range	Ipv6Address Range	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 4-34 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 4-35 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 4-36 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

OCNRF Access Token Service Usage Details

OCNRF implements Nnrf_AccessToken service (used for OAuth2 authorization), along with the "Client Credentials" authorization grant. It exposes a "Token Endpoint" where the Access Token Request service can be requested by NF Service Consumers.

The Nnrf_AccessToken service operation is defined as follows:

- Access Token Request (i.e. Nnrf_AccessToken_Get)

> **Note:**
>
> This procedure is specific to OCNRF Access Token service operation. OCNRF general configurations, database and database specific secret creation are not part of this procedure.

Procedure to use OCNRF Access Token Service Operation

This procedure provides step by step details which are needed to use 3GPP defined Access Token Service Operation supported by OCNRF.

1. **Create OCNRF private key and public certificate**
   This step explains need to create the OCNRF private keys and public certificates. Private key are used by OCNRF NF to sign the Access Token generated. It shall be available only with OCNRF. Public certificates are used by producer NFs to validate the access token generated by OCNRF. So, public certificates shall be available with producer network functions. Two types of signing algorithms are supported by OCNRF. For both types different keys and certificates required to be generated:

   - ES256: ECDSA digital signature with SHA-256 hash algorithm
   - RS256: RSA digital signature with SHA-256 hash algorithm

   Any one/both of algorithm files can be generated depending upon usage of hash algorithms. One algorithm depending upon configuration at OCNRF will decide which key will used to sign the Access Token.

   > **Note:**
   >
   > Creation process for private keys, certificates and passwords is on discretion of user/operator.

   Sample keys and certificates:
   After execution of this step, there will be private keys and public certificates of OCNRF (generated files depends upon algorithms chosen by operator/user).

   For example:

   ES256 based keys and certificates:

   - ecdsa_private_key.pem
   - ecdsa_certificate.crt

   RS256 based keys and certificates:

   - rsa_private_key.pem
   - rsa_certificate.crt

2. **Password to keep safely the generated keys and certificate inside OCNRF container**
   This step explains the create password that is used to keep safely the generated keys and certificate inside OCNRF container.

Sample step to create:

```
echo qwerpoiu > keystore_password.txt
```

where, `qwerpoiu` is the password and `keystore_password.txt` is the target password file

> **Note:**
>
> This file is provided in Kubernetes secret.

After execution of this step, file will be available with password.

For example: `keystore_password.txt`

3. **Name space creation for Secrets**
   This step explains the need for creating kubernetes namespace in which kubernetes secrets will be created for OCNRF private keys, OCNRF public certificate and keystore password. Refer to Creating OCNRF Namespace section in OCNRF Installation and Upgrade guide.

   > **Note:**
   >
   > - Different namespaces or same namespace can be used for OCNRF private keys, OCNRF public certificate and keystore password.
   > - Namespace(s) shall have RBAC resources defined with required privileges.
   > - It can be same namespace as for OCNRF.
   > - Namespace will be available in which required secrets can be created in next steps

4. **Secret creation for OCNRF private keys, OCNRF public certificate and keystore password**
   This step explain commands to create the kubernetes secret(s) in which OCNRF private keys, OCNRF public certificate and keystore password can be kept safely. Refer to Configuring Kubernetes Secret for Accessing OCNRF Database section in OCNRF Installation and Upgrade guide.

   > **Note:**
   >
   > Single secret can be created for OCNRF private keys, OCNRF public certificate and keystore password. Sample command is provided in steps to create single secret. In case, there is need to create separate secret for each entity, then same command can be used.

5. **Configure OCNRF custom_values.yaml with outcome details of Steps 1 to 4**
   This step explains customize the OCNRF custom_values.yaml to use the OCNRF private keys, OCNRF public certificate, keystore password file, secrets, and

secret namespace. Refer to Configuring Secret for Enabling AccessToken Service section in OCNRF Installation and Upgrade guide.

Key Attributes in OCNRF custom_values.yaml:

- nfaccesstoken.oauth.nrfInstanceId - OCNRF's NF Instance ID that will be used for signing AccessTokenClaim.
- nfaccesstoken.oauth.initialAlgorithm - Signing algorithm which will be used by Access Token microservice. This is default value.
- NF Access Token OCNRF Private Key Details
  a. k8SecretName - K8 Secret Name for OCNRF Access Token Private key
  b. k8NameSpace - Namespace for OCNRF Access Token Private key Secret
  c. rsa.filename - Key File name which is OCNRF Access Token Private Key for RSA algorithm
  d. ecdsa.filename - Key File name which is OCNRF Access Token Private Key for ECDSA algorithm
- NF Access Token OCNRF Public Certificate Details
  a. k8SecretName - K8 Secret Name for OCNRF Access Token Public Certificate
  b. k8NameSpace - Namespace for OCNRF Access Token Public Certificate Secret
  c. rsa.filename - Key File name which is OCNRF Access Token Public Certificate for RSA algorithm
  d. ecdsa.filename - Key File name which is OCNRF Access Token Public Certificate for ECDSA algorithm
- NF Access Token Key Store Password Details
  a. k8SecretName - K8 Secret Name for OCNRF Access Token Key Store Password
  b. k8NameSpace - Namespace for OCNRF Access Token Key Store password Secret
  c. filename - KeyStore password file

5

Configuring OCNRF using CNC Console

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

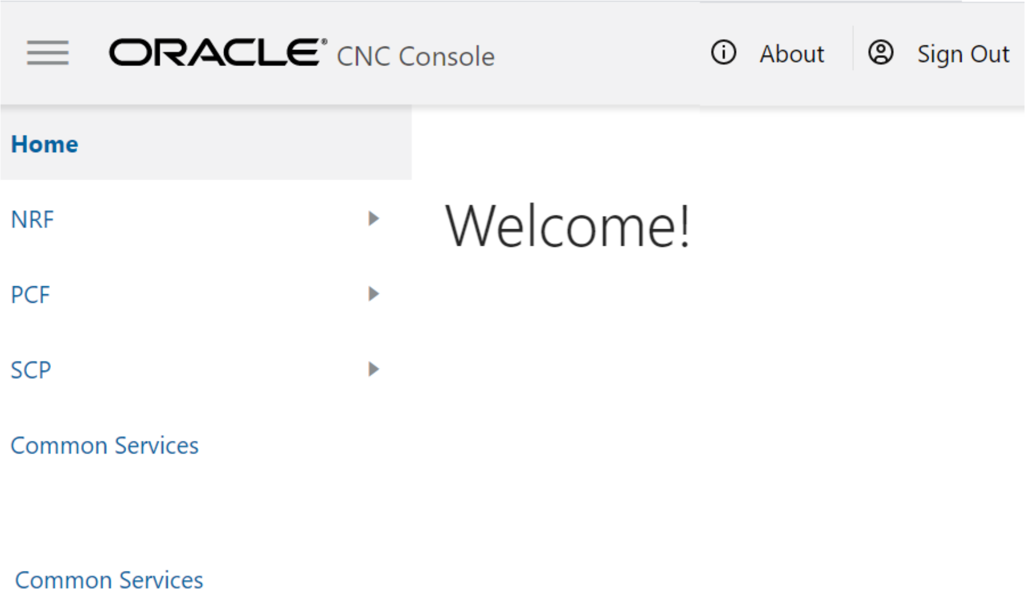
CNC Console Interface

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 > Screening Rules > CALLBACK URI**. Select **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. Enter the values for the attributes as per the requirement. Refer to Configuring NF Screening for more information in parameter values and description.
4. Click **Save**.

Modifying NF Callback URIs

The user can add, edit and delete the NF Callback URIs.

Adding the NF Callback URIs

To add a NF Callback URIs:

1. Click **Add** from the top left of the NF Callback URI table. The **Add NF Callback URI** Screen appears.
2. Enter the attribute values. Refer to Configuring NF Screening for more information in parameter values and description.
3. Click **Save**.

Editing the NF Callback URIs

To edit an existing NF Callback URIs:

1. Click **Edit** from the top left of the NF Callback URI table. The **Edit NF Callback URI** Screen appears.
2. Enter the attribute values.
3. Click **Save**.

Deleting the NF Callback URIs Parameters

To delete a NF Callback URIs Parameter:

1. Click **Delete** from the action items of NF Callback URIs Screen.
   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

To configure NF FQDN parameters follow the procedure:

1. From the left navigation menu, navigate to **NRF > Screening Rules > NF FQDN**. Select **FQDN**.
2. Click **Edit** from the top right side to edit or update a NF FQDN parameter. The screen is enabled for modification.
3. Enter the values for the attributes as per the requirement. Refer to Configuring NF Screening for more information in parameter values and description.
4. Click **Save**.

Modifying NF FQDN

The user can add, edit or delete the NF FQDN.

Adding the NF FQDN

To add a NF FQDN:

1. Click **Add** from the top left of the NF FQDN table. The **Add NF FQDN** Screen appears.
2. Enter the attribute values. Refer to Configuring NF Screening for more information in parameter values and description.
3. Click **Save**.

Editing the NF FQDN

To edit an existing NF FQDN:

1. Click **Edit** from the top left of the NF FQDN table. The **Edit NF FQDN** Screen appears.
2. Enter the attribute values.
3. Click **Save**.

Deleting the NF FQDN Parameters

To delete a NF FQDN Parameter:

1. Click **Delete** from the action items of NF FQDN Screen.
   The *" Do you want to delete the record"* message appears.
2. Click **OK** to delete the parameter.

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 > Screening Rules > NF IP Endpoint**. Select **NF IP Endpoint**.
2. Click **Edit** from the top right side to edit or update a NF IP Endpoint parameters. The screen is enabled for modification.
3. Enter the values for the attributes as per the requirement. Refer to Configuring NF Screening for more information in parameter values and description.
4. Click **Save**.

Modifying NF IP Endpoint

The user can add, edit or delete the NF IP Endpoint.

Adding the NF IP Endpoint

To add a NF IP Endpoint:

1. Click **Add** from the top left of the NF IP Endpoint table. The **Add NF IP Endpoint** Screen appears.
2. Enter the attribute values. Refer to Configuring NF Screening for more information in parameter values and description.
3. Click **Save**.

Editing the NF IP Endpoint

To edit an existing NF IP Endpoint:

1. Click **Edit** from the top left of the NF IP Endpoint table. The **Edit NF IP Endpoint** Screen appears.
2. Enter the attribute values. Refer to Configuring NF Screening for more information in parameter values and description.
3. Click **Save**.

Deleting the NF IP Endpoint Parameters

To delete a NF IP Endpoint Parameters:

1. Click **Delete** from the action items of NF IP Endpoint Screen.
   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 > Screening Rules > NF IP Type Register**. Select **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. Enter the values for the attributes as per the requirement. Refer to Configuring NF Screening for more information in parameter values and description.
4. Click **Save**.

Modifying NF IP Type Register

The user can add, edit or delete the NF IP Type Register.

Adding the NF IP Type Register

To add a NF IP Type Register:

1. Click **Add** from the top left of the NF IP Type Register table. The **Add NF IP Type Register** Screen appears.
2. Enter the attribute values. Refer to Configuring NF Screening for more information in parameter values and description.
3. Click **Save**.

Editing the NF IP Type Register

To edit an existing NF IP Type Register:

1. Click **Edit** from the top left of the NF IP Type Register table. The **Edit NF IP Type Register** Screen appears.
2. Enter the attribute values.
3. Click **Save**.

Deleting the NF IP Type Register Parameters

To delete a NF IP Type Register Parameters:

1. Click **Delete** from the action items of NF IP Type Register Screen.
   The *" Do you want to delete the record"* message appears.
2. Click **OK** to delete the parameter.

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 > Screening Rules > NF IP Type Register PLMN ID**. Select **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. Enter the values for the attributes as per the requirement. Refer to Configuring NF Screening for more information in parameter values and description.
4. Click **Save**.

Modifying PLMN ID

The user can add, edit or delete the PLMN ID.

Adding the PLMN ID

To add a PLMN ID:

1. Click **Add** from the top left of the PLMN ID table. The **Add PLMN ID** Screen appears.
2. Enter the attribute values. Refer to NRF User's Guide for more information in attribute values and description.

3. Click **Save**.

Editing the PLMN ID

To edit an existing PLMN ID:

1. Click **Edit** from the top left of the PLMN ID table. The **Edit PLMN ID** Screen appears.
2. Enter the attribute values.
3. Click **Save**.

Deleting the PLMN ID Parameters

To delete a PLMN ID Parameters:

1. Click **Delete** from the action items of PLMN ID Screen.
   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 > Screening Rules > System Options**. Select **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 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. The **Add** Screen appears.
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 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. Enter the attribute values.
4. Click **Save**.

Deleting Configuration list

To delete a Configuration list:

1. Click **Edit** from the top left 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	Notes
1	Total number of ingress requests	Total number of requests received at OCNRF	apigateway_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	Notes
2	NF Register Success	Total number of successful NFRegister service operations at OCNRF	apigateway_http_responses_total{Status=\"201 CREATED\",Uri=~\".*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	apigateway_http_responses_total{Status=\"200 OK\",Uri=~\".*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	Notes
4	NF Update Success (Partial Replacement)	Total number of successful NFUpdate service operations at OCNRF	apigateway_http_responses_total{Status=~\".*2.*\",Uri=~\".*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	apigateway_http_responses_total{Status=~\".*2.*\",Uri=~\".*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	Notes
6	Access Token Success	Total number of successful Access Token service operations at OCNRF	apigateway_http_responses_total{Status=\"200 OK\",Uri=~\".*/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	apigateway_http_responses_total{Status=\"204 NO_CONTENT\",Uri=~\".*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	Notes
8	NF Subscribe Success	Total number of successful NFSubscribe service operations at OCNRF	apigateway_http_responses_total{Status=\"201 CREATED\",Uri=~\".*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	apigateway_http_responses_total{Status=\"204 NO_CONTENT\",Uri=~\".*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	Notes
10	NF Discover Success	Total number of successful NFDiscover service operations at OCNRF	apigateway_http_responses_total{Status=~\"2.*\",Uri=~\".*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)	apigateway_http_responses_total{Status=~"4.*", Uri=~".*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	Notes
12	4xx Responses (Subscriptions)	Total number of 4xx responses (NfSubscribe/ NfUnsubscribe)	apigateway_http_responses_total{Status=~"4.*", Uri=~".*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)	apigateway_http_responses_total{Status=~"4.*", Uri=~".*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	Notes
14	4xx Responses (AccessToken)	Total number of 4xx responses(NfAccessToken)	apigateway_http_responses_total{Status=~"4.*", Uri=~".*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)	apigateway_http_responses_total{Status=~"5.*", Uri=~".*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	Notes
16	5xx Responses (Subscriptions)	Total number of 5xx responses (NfSubscribe/NfUnsubscribe)	apigateway_http_responses_total{Status=~"5.*", Uri=~".*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)	apigateway_http_responses_total{Status=~"5.*", Uri=~".*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	Notes
18	5xx Responses (AccessToken)	Total number of 5xx responses(NfAccessToken)	apigateway_http_responses_total{Status=~"5.*", Uri=~".*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	

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions	Notes
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	
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_requests_total	NfUpdates Responses Total	NrfLevel, NfInstanceId , RequesterNfType, UpdateType(Partial/Complete), HttpStatusCode	

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions	Notes
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 Responses 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	
28	NF De-Registration Responses Total	Number of De-registration responses sent	ocnrf_nfDeregister_tx_responses_total		NrfLevel, NfInstanceId, RequesterNfType, HttpStatusCode	

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions	Notes
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, NfType, NfInstanceId	
35	NF Status Subscribe Requests Total	Number of NStatusSubscribe requests received	ocnrf_nfStatusSubscribe_rx_requests_total		NrfLevel, RequesterNfType, OperationType	

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions	Notes
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	

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions	Notes
41	NF Status Notifications Requests Failed	Number of NfStatusNotify requests failed to sent out ocnrf_nfStatusNotify_requests_failed_total	ocnrf_nfStatusNotify_requests_failed_total		NrfLevel, NotificationEventType, TargetNfType	This metric indicates whether Notification message failed to sent out from OCNRF (including Egress gateway too)
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	

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions	Notes
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_count_total	Discovered profiles [TargetNfType :- {{TargetNfType}}, Bucket :- {{ Bucket }}]	NrfLevel, TargetNfType, BucketSize	
46	Active Registrations	Number of active registered NFs at any point of time	ocnrf_active_registrations_count_total	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	Notes
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)	ocnrf_message_processing_time_seconds	Avg NRF Latency {{ ServiceOperation }} {{ RequesterNfType }}	NrfLevel, RequesterNfType, ServiceOperation	Latency calculated by this metric doesn't include time taken by OCNRF API gateway.
48	Avg NRF Latency	Time (in microseconds) to process an ingress request. Measured from when the request is received to when the response is sent	apigateway_request_latency_seconds	Avg NRF Latency		
49	OCNRF database operations	Database operation count corresponding to every service operation	ocnrf_dbmetric_total	Method, DBOperation, NrfLevel, HttpStatusCode		

Table 6-2 (Cont.) OCNRF Metrics

Sl. No#	Metric Name	Metric Details	Metric filter	Recommended legend to see dimension level data (as applicable)	Dimensions	Notes
50	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.

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

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions	Notes
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.
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.

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

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions	Notes
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, ServiceName, NrfLevel
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, ServiceName, NrfLevel, HttpStatusCode

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

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
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, ServiceName, 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, ServiceName, 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, ServiceName, NrfLevel, RejectionReason HttpStatusCode RejectionReason = ProducerWithRequestedNfInstanceIdNotFound
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, ServiceName, NrfLevel, RejectionReason HttpStatusCode RejectionReason = InconsistentScope

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

Sl. No#	Metric Name	Metric Details	Metric filter	Service Operation	Dimensions
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, ServiceName, 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, ServiceName, 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, ServiceName, NrfLevel, RejectionReason HttpStatusCode RejectionReason = ProducerNFTypeMismatch

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, ServiceName, 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, ServiceName, 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_nfSubscription_expired_total		NrfLevel, SubscriptionId

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. Please view the Oracle Help Center site for the information about OCNRF Custom Templates.

Table 6-8 KPI Details

Sl. No#	KPI Name	KPI Details	Metric used for KPI	Service Operation	Response code
1	OCNRF Ingress Request	Rate of HTTP requestes recieved at OCNRF Ingress Gateway	apigateway_http_requests_total	All	Not Applicable
2	NF Register Success		sum(increase(apigateway_http_responses_total{Status="201 CREATED",Uri=~".*nnrf-nfm/v1/nf-instances.*",Method="PUT"}[5m]))	NFRegister	201
3	NF Update Success (Complete Replacement)		sum(increase(apigateway_http_responses_total{Status="200 OK",Uri=~".*nnrf-nfm/v1/nf-instances.*",Method="PUT"}[5m]))	NFUpdate	200

Table 6-8 (Cont.) KPI Details

Sl. No#	KPI Name	KPI Details	Metric used for KPI	Service Operation	Response code
4	NF DeRegister Success		sum(increase(apigateway_http_responses_total{Status="204 NO_CONTENT",Uri=~".*nnrf-nfm/v1/nf-instances.*",Method="DELETE"}[5m]))	NFDeregister	204
5	NF Subscribe Success		sum(increase(apigateway_http_responses_total{Status="201 CREATED",Uri=~".*nnrf-nfm/v1/subscriptions.*",Method="POST"}[5m]))	NFStatusSubscribe	201
6	NF Unsubscribe Success		sum(increase(apigateway_http_responses_total{Status="204 NO_CONTENT",Uri=~".*nnrf-nfm/v1/subscriptions.*",Method="DELETE"}[5m]))	NFStatusUnsubscribe	204
7	NF Discover Success		sum(increase(apigateway_http_responses_total{Status=~"2.*",Uri=~".*nnrf-disc/v1/nf-instances.*",Method="GET"}[5m]))	NFDiscover	200
8	4xx Responses (NF-Instances)		sum(increase(apigateway_http_responses_total{Status=~"4.*",Uri=~".*nnrf-nfm/v1/nf-instances.*"}[5m]))	NFRegister/NFUpdate/NFDeregister	4xx
9	4xx Responses (Subscriptions)		sum(increase(apigateway_http_responses_total{Status=~"4.*",Uri=~".*nnrf-nfm/v1/subscriptions.*"}[5m]))	NFStatusSubscribe/NFStatusUnsubscribe	4xx
10	4xx Responses (Discovery)		sum(increase(apigateway_http_responses_total{Status=~"4.*",Uri=~".*nnrf-disc/v1/nf-instances.*"}[5m]))	NFDiscover	4xx
11	5xx Responses (NF-Instances)		sum(increase(apigateway_http_responses_total{Status=~"5.*",Uri=~".*nnrf-nfm/v1/nf-instances.*"}[5m]))	NFRegister/NFUpdate/NFDeregister	5xx

Table 6-8 (Cont.) KPI Details

Sl. No#	KPI Name	KPI Details	Metric used for KPI	Service Operation	Response code
12	5xx Responses (Subscriptions)		asum(increase(apigateway_http_responses_total{Status=~"5.*",Uri=~".*nnrf-nfm/v1/subscriptions.*"}[5m]))	NFStatusSubscribe/ NFStatusUnsubscribe	5xx
13	5xx Responses (Discovery)		sum(increase(apigateway_http_responses_total{Status=~"5.*",Uri=~".*nnrf-disc/v1/nf-instances.*"}[5m]))	NFDiscover	5xx

OCNRF Alerts

This section includes information about alerts for OCNRF.

Table 6-9 OCNRF Alert Details

Alert Name	Alert Description	Severity	Alert Details
OcnrfTrafficRateAboveCriticalThreshold	Alert if Ingress traffic reaches 95% of Max requests per second	Critical	Traffic Rate is above critical threshold.
OcnrfTrafficRateAboveMajorThreshold	Alert if Ingress traffic reaches 90% of Max requests per second	Major	Traffic Rate is above major threshold.
OcnrfTrafficRateAboveMinorThreshold	Alert if Ingress traffic reaches 80% of Max requests per second	Minor	Traffic Rate is above minor threshold.
OcnrfTransactionErrorRateAbove0.1Percent	Alert if error rate exceeds 0.1% of the total transactions	Warning	Transaction Error rate is above 0.1 Percent of Total Transactions.
OcnrfTransactionErrorRateAbove1Percent	Alert if error rate exceeds 1% of the total transactions	Warning	Transaction Error rate is above 1 Percent of Total Transactions.
OcnrfTransactionErrorRateAbove10Percent	Alert if error rate exceeds 10% of the total transactions	Minor	Transaction Error rate is above 10 Percent of Total Transactions.
OcnrfTransactionErrorRateAbove25Percent	Alert if error rate exceeds 25% of the total transactions	Major	Transaction Error rate is above 25 Percent of Total Transactions.

Table 6-9 (Cont.) OCNRF Alert Details

Alert Name	Alert Description	Severity	Alert Details
OcnrfTransactionErrorRateAbove50Percent	Alert if error rate exceeds 50% of the total transactions	Critical	Transaction Error rate is above 50 Percent of Total Transactions.
OcnrfRegisteredNFsBelowThreshold	Alert if the number of registered NFs is approaching minor threshold (The operator shall define the threshold as per requirement. Default range: 20-29)	Warning	The number of registered NFs is approaching minor threshold. **Note**: The threshold ranges needs to be updated accordingly to the requirement.
OcnrfRegisteredNFsBelowMinorThreshold	Alert if the number of registered NFs is below minor threshold (The operator shall define the threshold as per requirement. Default range: 10-19)	Minor	The number of registered NFs is below minor threshold. **Note**: The threshold ranges needs to be updated accordingly to the requirement.
OcnrfRegisteredNFsBelowMajorThreshold	Alert if the number of registered NFs is below major threshold (The operator shall define the threshold as per requirement. Default range: 2-9)	Major	The number of registered NFs is below major threshold. **Note**: The threshold ranges needs to be updated accordingly to the requirement.
OcnrfRegisteredNFsBelowCriticalThreshold	Alert if the number of registered NFs is below critical threshold (The operator shall define the threshold as per requirement. Default range: < 2)	Critical	The number of registered NFs is below critical threshold. **Note**: The threshold ranges needs to be updated accordingly to the requirement.
OcnrfDbReplicationStatusInactive	The alert raised when database replication is inactive.	Critical	The Database Replication Status is currently INACTIVE.

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

OCNRF Alert Configuration

Follow the steps below for OCNRF Alert configuration in Prometheus:

> **Note:**
>
> 1. By default `Namespace` for OCNRF is `ocnrf` that must be update as per the deployment.
> 2. The `ocnrf-config-1.7.2.0.0.zip` file can be downloaded from OHC. Unzip the `ocnrf-config-1.7.2.0.0.zip` package after downloading to get `NrfAlertrules-1.7.2.yaml` file.
>
> > **Note:**
> >
> > Alert file is packaged along with OCNRF Custom Templates. Download the file from OHC. Refer to OCNRF Installation and Upgrade guide for more details.

Procedure

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.

OCNRF Alert Config Details

> **Note:**
>
> - By default the NameSpace is set to ***ocnrf***. Update it according to the requirement.
> - Update the number of registered NFs according to the requirement.
> - Max request/sec in consideration is 1000 requests /second

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 >> 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-1.7.2.mib
  This is considered as OCNRF top level mib file, where the Objects and their data types are defined.
- OCNRF-MIB-1.7.2.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.