

Deploying and Configuring NRF

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- Deploying and Accessing NRF, on page 2
- Loading Day 1 Configuration, on page 8

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	5G-NRF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	2026.01

Feature Description

The NRF deployment and configuration procedure involves deploying the NRF through the Subscriber Microservices Infrastructure (SMI) Cluster Deployer and configuring the settings or customizations through the NRF Operations (Ops) Center. The Ops Center is based on the ConfD CLI. The NRF configuration includes the NRF profile data configuration and the externally visible IP addresses and ports.

NRF Ops Center

The Ops Center is a system-level infrastructure that provides the following functionality:

- A user interface to trigger a deployment of micro-services with the flexibility of providing variable helm chart parameters to control the scale and properties of Kubernetes objects (deployment, pod, services, and so on) associated with the deployment.
- A user interface to push application specific configuration to one or more micro-services through Kubernetes configuration maps.
- A user interface to issue application-specific execution commands (such as, show commands and clear commands). These commands:
  - invoke some APIs in application-specific pods
  - display the information returned by the application on the user interface

NRF uses the Ops Center infrastructure to deploy its helm charts. The NRF Ops Center has its own `render.yaml` file (helm template file).

Prerequisites

Before deploying NRF on the SMI layer:

- Make sure that all the virtual network functions (VNFs) are deployed.
- Run the SMI synchronization operation for the NRF Ops Center and Cloud Native Common Execution Environment (CN-CEE).

Deploying and Accessing NRF

This section describes how to deploy NRF and access the NRF Ops Center.

Deploying NRF

The SMI platform is responsible for deploying and managing the Cloud Native 5G NRF application and other network functions.

For deploying NRF Ops Center on a vCenter environment, see *Deploying and Upgrading the Product* section in the *UCC SMI Cluster Deployer Operations Guide*.

For deploying NRF Ops Center on a OpenStack environment, see *UAME-based VNF Deployment* section in the *UAME-based 4G and 5G VNF Deployment Automation Guide, Release 6.9*

Accessing the NRF Ops Center

You can connect to the NRF Ops Center through SSH or the web-based CLI console.

- SSH:

  ssh admin*@ops_center_pod_ip* **-p 2024**

- Web-based console:

  1. Log in to the Kubernetes master node.
  2. Run the following command:

     kubectl get ingress <*namespace*>

     The available ingress connections get listed.
  3. Select the appropriate ingress and access the NRF Ops Center.
  4. Access the following URL from your web browser:

     cli.<*namespace*>**-ops-center**.<*ip_address*>.**nip.io**

By default, the Day 0 configuration is loaded into the NRF.

Day 0 Configuration

To view the Day 0 configuration, run the following command:

```
show running-config
```

The following code is a sample Day 0 configuration:

```
helm default-repository base-repos
helm repository base-repos
 url https://charts.172.29.200.5.nip.io/nrf.2020.04.0.i35
exit
k8s name          smi-5g-3
k8s namespace     nrf-perf
k8s nf-name       nrf
k8s registry      docker.172.29.200.5.nip.io/nrf.2020.04.0.i35
k8s single-node   false
k8s use-volume-claims false
k8s ingress-host-name 10.86.62.122.nip.io
aaa authentication users user admin
 uid        1117
 gid        1117
 password   $1$Id/Bm4st$enxcWXA8KG7RniInJLwki1
 ssh_keydir /tmp/admin/.ssh
 homedir    /tmp/admin
exit
aaa ios level 0
 prompt "\h> "
exit
aaa ios level 15
 prompt "\h# "
exit
```

```
aaa ios privilege exec
 level 0
  command action
  exit
  command autowizard
  exit
  command enable
  exit
  command exit
  exit
  command help
  exit
  command startup
  exit
 exit
 level 15
  command configure
  exit
 exit
exit
nacm write-default deny
nacm groups group admin
 user-name [ admin ]
exit
nacm rule-list admin
 group [ admin ]
 rule any-access
  action permit
 exit
exit
nacm rule-list confd-api-manager
 group [ confd-api-manager ]
 rule any-access
  action permit
 exit
exit
nacm rule-list ops-center-security
 group [ * ]
 rule change-self-password
  module-name       ops-center-security
  path              /smiuser/change-self-password
  access-operations exec
  action            permit
 exit
 rule smiuser
  module-name       ops-center-security
  path              /smiuser
  access-operations exec
  action            deny
 exit
exit
```

Configuring the NRF Rest Endpoint

To configure the IP address and port of the NRF rest endpoint, use the following sample configuration:

```
config
   cdl datastore datastore_name
      endpoint sbi
         interface { auth | disc | mgmt }
            loopbackPort port_number
```

```
        api-root api_string
      exit
```

NOTES:

- **endpoint sbi**: Specifiy the service-based interface (SBI) as the endpoint.
- **interface { auth | disc | mgmt }**: Specify the NRF authentication, NRF discovery, or NRF management interface.
- **loopbackPort** *port_number*: Specify the internal port number of the loopback host. The NRF uses this port for NF status notification.
- **api-root** *api_string*: Specify the API prefix of the deployment-specific service that is used within { apiRoot }.

Configuration Example

The following is an example configuration.

```
config
 cdl datastore session
  endpoint sbi
   interface mgmt
    loopbackPort 8094
    api-root  root
    exit
   interface disc
    loopbackPort 8095
    api-root  root
    exit
```

Configuring the NRF Service Engine

To configure the NRF service engine, use the following sample configuration:

```
config
   cdl datastore datastore_name
      endpoint service
         nodes nodes_number
         replicas replicas_number
         exit
      exit
   k8 nrf local datastore-endpoint datastore_ep
   nrf-profile instance-id nrf_instance_id
   nrf-profile plmn-ids hplmn_id
      is-home { false | true }
      mcc mcc_value
      mnc mnc_value
      exit
  nrf-profile profile-settings { nf-heartbeat-timer-seconds heartbeat_timer
 | nf-discovery-validity-period validity_period |
nf-heartbeat-grace-time-seconds grace_time |
```

```
suspended-nf-profile-purge-time-mins suspended_time }
  end
```

NOTES:

- **nodes** *nodes_number*: Specifiy the number of nodes for the service endpoint. To deploy the pods on different worker nodes, set the **nodes** parameter.
- **replicas** *replicas_number*: Specify the number of instances for the service endpoint.
- **k8 nrf local datastore-endpoint** *datastore_ep*: Specify the datastore endpoint for NRF local configuration.
- **nrf-profile instance-id** *nrf_instance_id*: Specify the NRF instance ID.
- **nrf-profile plmn-ids** *plmn_id*: Specify the NRF Home PLMN ID.
- **is-home { false | true }**: Specify whether it is home PLMN or not. Default value: **false**.
- **mcc** *mcc_value*: Specify the PLMN ID Mobile Country Code (MCC) as a 3-digit integer.
- **mnc** *mnc_value*: Specify the PLMN ID Mobile Network Code (MNC) as a 2- or 3-digit integer.
- **nrf-profile profile-settings { nf-heartbeat-timer-seconds** *heartbeat_timer* **| nf-discovery-validity-period** *time_period* **| nf-heartbeat-grace-time-seconds** *heartbeat_grace_time* **| suspended-nf-profile-purge-time-mins** *suspended_time_period* **}**: Specify the NRF NF profile settings.
  - **nf-heartbeat-timer-seconds** *heartbeat_timer*: Specify the time in seconds after which NF should update NRF if it is alive. *heartbeat_timer* must be an integer in the range of 1 to 2592000. Default value: 300 seconds.
  - **nf-discovery-validity-period** *validity_period*: Specify the time in minutes after which an NF should do discovery again. *validity_period* must be an integer in the range of 30 to 1440. Default Value: 180 minutes.
  - **nf-heartbeat-grace-time-seconds** *grace_time*: Specify the time in seconds after which NRF would mark the NF instance SUSPENDED if NF does not update NRF. *grace_time* must be an integer in the range of 0 to 86400. Default value: 300 seconds.
  - **suspended-nf-profile-purge-time-mins** *suspended_time*: Specify the time in minutes after which a suspended NF profile will be removed from the NRF profile database. *suspended_time* must be an integer in the range of 5 to 43200. Default value: 1440 minutes.

Configuration Example

The following is an example configuration.

```
config
 endpoint service
  nodes    3
  replicas 2
  exit
 k8 nrf local datastore-endpoint datastore-ep-session:8882
 nrf-profile instance-id      350f9d40-7c99-412e-bea1-5db52f7c0bf4
 nrf-profile plmn-ids hplmn
  is-home true
  mcc     123
  mnc     456
  exit
 nrf-profile plmn-ids roam-plmn
```

```
 is-home false
 mcc     789
 mnc     45
 exit
nrf-profile profile-settings nf-heartbeat-timer-seconds 300
nrf-profile profile-settings nf-discovery-validity-period 12
nrf-profile profile-settings nf-heartbeat-grace-time-seconds 300
nrf-profile profile-settings suspended-nf-profile-purge-time-mins 5
```

Configuring the Datastore Services

To configure the replica and shard count for the profile database and subscription database, use the following sample configuration:

```
config
   cdl zookeeper replica replica_number
   cdl datastore datastore_name
      index { map map_value | write-factor write_factor }
      slot { replica replica_number | map map_shards_number | write-factor
write_factor }
      exit
   cdl kafka replica replica_number
   k8 label cdl-layer key label_key value label_value
   exit
```

NOTES:

- **cdl zookeeper replica** *replica_number*: Specify the number of HA instances to be created for the CDL zookeeper configuration.
- **cdl datastore** *datastore_name*: Configure the CDL datastore.
- **index { map** *map_value* **| write-factor** *write_factor* **}**: Specify the configuration for index pods.
  - **map** *map_value*: Specifiy the number of partitions to be created for an index. *map_value* must be an integer in the range of 1–1024. Default value: 1.
  - **write-factor** *write_factor*: Specify the number of copies to be written into an index before successful response. *write_factor* must be an integer in the range of 0–2. Default value: 1.

Note This keyword is deprecated.

- **slot { replica** *replica_number* **| map** *map_shards_number* **| write-factor** *write_factor* **}**: Specify the configuration for slot pods.
  - **replica** *replica_number*: Specify the number of HA instances to be created in a slot. *replica_number* must be an integer in the range of 1–4. Default value: 1.
  - **map** *map_shards_number*: Specify the number of partitions to be created in a slot. *map_shards_number* must be an integer in the range of 1–1024. Default value: 1.
  - **write-factor** *write_factor*: Specify the number of copies to be written into a slot before successful response. *write_factor* must be an integer in the range of 0–16. Ensure that the value is lower than or equal to the number of replicas. Default value: 1.

- **cdl kafka replica** *replica_number*: Specify the number of kafka replicas. *replica_number* must be an integer in the range of 1–16. Default value: 2.
- **k8 label cdl-layer key** *label_key* **value** *label_value*: Specify the K8 CDL layer configuration.

Configuration Example

The following is an example configuration.

```
config
  cdl zookeeper replica 1
  cdl datastore session
    index map 1
    index write-factor 1
    slot replica 1
    slot map 1
    slot write-factor 1
    exit
  cdl kafka replica 1
  k8 label cdl-layer key disktype value ssd
  exit
```

Configuring the Logging Service

To configure the logging service, use the following sample configuration:

```
config
  logging level { application | monitor-subscriber | tracing | transaction
 } { trace | debug | info | warn | error | off }
  exit
```

NOTES:

- **logging level { application | monitor-subscriber | tracing | transaction } { trace | debug | info | warn | error | off }**: Specify the logging level service and configure the logging level.

Loading Day 1 Configuration

To load the Day 1 configuration for NRF, run the following command:

```
ssh admin@ops_center_pod_ip -p 2024  < Day1config.cli
```



Note The Day1config.cli file contains the necessary parameters that are required for the Day 1 configuration.

Alternatively, you can copy the configuration and paste it in the NRF Ops Center CLI to load the Day 1 configuration.

```
config
        <Paste the Day 1 configuration here>
   commit
   exit
```

A sample Day1config.cli file, which contains the Day 1 configuration for NRF is as follows:

```
cdl node-type session
cdl zookeeper replica 3
cdl logging default-log-level error
cdl datastore session
 slice-names [ slice1 ]
 endpoint replica 2
 endpoint copies-per-node 2
 endpoint settings index-timeout-ms 500
 endpoint settings slot-timeout-ms 1000
 index replica 2
 index map    1
 index write-factor 1
 slot replica 2
 slot map     4
 slot write-factor 2
exit
cdl kafka replica 2
cdl kafka storage 1
etcd replicas 3
instance instance-id 1
 endpoint service
  replicas 2
  nodes    3
 exit
 endpoint sbi
  replicas 2
  nodes    3
  vip-ip 1.1.1.1
  interface mgmt
   loopbackPort 8094
   api-root     root
  exit
  interface disc
   loopbackPort 8095
   api-root     root
  exit
  interface auth
  loopbackPort 8096
  exit
 exit
exit
logging level application error
logging level transaction error
logging level tracing error
deployment
 app-name     NRF
 cluster-name cluster-nrf
 dc-name      bxb
exit
k8 nrf local etcd endpoint host 2379
k8 nrf local datastore-endpoint datastore-ep-session:8882
k8 label protocol-layer key smi.cisco.com/node-type value proto
exit
k8 label service-layer key smi.cisco.com/node-type value svc
exit
k8 label cdl-layer key smi.cisco.com/node-type value session
exit
k8 label oam-layer key smi.cisco.com/node-type value oam
exit
instances instance 1
 system-id  NRF
 cluster-id NRF
```

```
 slice-name slice1
exit
local-instance instance 1
nrf-profile instance-id 350f9d40-7c99-412e-bea1-5db52f7c0bf3
nrf-profile locality Bangalore
nrf-profile plmn-ids nfplmn
 is-home true
 mcc     310
 mnc     12
exit
nrf-profile plmn-ids plmn-1
 is-home true
 mcc     404
 mnc     31
exit
nrf-profile profile-settings nf-heartbeat-timer-seconds 3600
nrf-profile profile-settings override-nf-heartbeat-timer true
nrf-profile profile-settings nf-discovery-validity-period 200
nrf-profile profile-settings nf-heartbeat-grace-time-seconds 1800
nrf-profile profile-settings nf-subscription-validity-period 3600
nrf-profile profile-settings nf-subscription-cache-timer 60
nrf-profile db-non-unique-indexes AMF_Discovery
 fields TARGET_NF_TYPE
  condition MATCH
 exit
 fields TAI
  condition SUBSTRING_MATCH
 exit
 fields SNSSAIS
  condition SUBSTRING_MATCH
 exit
 fields TARGET_PLMNS
  condition SUBSTRING_MATCH
 exit
exit
nrf-profile db-non-unique-indexes LR_NfType
 fields TARGET_NF_TYPE
  condition MATCH
 exit
exit
nrf-profile db-non-unique-indexes PCF_Discovery
 fields TARGET_NF_TYPE
  condition MATCH
 exit
 fields TARGET_PLMNS
  condition SUBSTRING_MATCH
 exit
exit
nrf-profile db-non-unique-indexes SMF_Discovery
 fields TARGET_NF_TYPE
  condition MATCH
 exit
 fields DNN
  condition SUBSTRING_MATCH
 exit
 fields TARGET_PLMNS
  condition SUBSTRING_MATCH
 exit
exit
system mode running
```



Note In non-standard deployments that does not have the recommended dimensioning for the VMs, use the CLI configuration of **`cdl deployment-model small`**.