Oracle® Communications
Network Slice Selection Function (NSSF) Cloud Native Installation Guide

Release 1.3

F31725-03

May 2020

ORACLE®

Oracle Communications Network Slice Selection Function (NSSF) Cloud Native Installation Guide, Release 1.3

F31725-03

Contents

List of Tables

1
What's New in This Guide

New and Updated Features and their Configurations in Release 1.3

The helm parameters and the pre-deployment configuration of following functionalities:

- **Open Authorization (OAuth)** authorization
- **Rate Limiting**
- **Hypertext Transfer Protocol Secure (https)** protocol
- **Ingress Gateway/Egress Gateway**

For the pre-deployment configuration details, refer **NSSF Pre-deployment Configuration**.

For helm parameter details, refer Configuration Options During Deployment.

For functionality details, refer *Network Slice Selection Function (NSSF) Cloud Native User's Guide.*

2 My Oracle Support (MOS)

MOS (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support (CAS) can assist you with MOS registration.

Call the CAS 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 MOS, Select 2

You will be connected to a live agent who can assist you with MOS registration and opening a support ticket.

MOS is available 24 hours a day, 7 days a week, 365 days a year.

3 Introduction

This document includes information about NSSF installation in 5G environment.

Network Slice Selection Function (NSSF) selects the network slicing instance (NSI), determines the allowed network slice selection assistance information (NSSAI) and sets AMF to serve the User Equipment (UE). AMF can retrieve NRF, NSI ID, and target AMFs as part of UE initial registration and PDU establishment procedure.

Oracle NSSF interaction with NRF allows retrieving specific NF services to be used for the registration request. It also allows mechanism for registration and subsequent notification Function Instance Discovery.

The NSSF supports the following services:

- Nnssf_NSSelection Service
- Nnssf_NSSAIAvailability Service

NSSF References

- Cloud Native Environment (OC-CNE) Installation Guide
- Network Slice Selection Function (NSSF) Cloud Native User's Guide

Acronyms and Terminologies

The following table provides information about the acronyms and terminologies used within the document:

Table 3-1 Acronyms and Terminologies

Field	Description
AMF	Access and Mobility Management Function
Allowed NSSAI	NSSAI provided by the Serving PLMN during e.g. a Registration procedure, indicating the S-NSSAIs values the UE could use in the Serving PLMN for the current registration area.
Configured NSSAI	NSSAI provisioned in the UE applicable to one or more PLMNs.
OHC	Oracle Help Center
OSDC	Oracle Software Delivery Cloud
NEF	Network Exposure Function
Network Slice	A logical network that provides specific network capabilities and network characteristics.
NF	Network Function
NF service	A functionality exposed by a NF through a service based interface and consumed by other authorized NFs.

Table 3-1 (Cont.) Acronyms and Terminologies

Field	Description
NRF	Network Repository Function
NSI	Network Slice instance. A set of Network Function instances and the required resources (such as compute, storage and networking resources) which form a deployed Network Slice.
NSI ID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
NSSP	Network Slice Selection Policy
PLMN	Public Land Mobile Network
Requested NSSAI	NSSAI provided by the UE to the Serving PLMN during registration.
SD	Slice Differentiator
SEPP	Security Edge Protection Proxy
S-NSSAI	Single Network Slice Selection Assistance Information
SST	Slice/Service type

4

Installing NSSF

This section describes how to install NSSF on a cloud native environment. It contains the following topics:

- NSSF Prerequisites
- NSSF Pre-deployment Configuration
- NSSF Installation Sequence
- NSSF Installation Preparation
- NSSF Installation

NSSF Prerequisites

This section includes information about the necessary prerequisites for NSSF deployment.

Following are the prerequisites to install and configure NSSF:

NSSF Software

The NSSF software includes:

- NSSF Helm charts
- NSSF docker images

The following software must be installed:

Software	Version
Kubernetes	v1.12.5
HELM	v2.11.0

Additional software that needs to be deployed as per the requirement of the services:

Software	Chart Version
elasticsearch	1.21.1
elastic-curator	1.2.1
elastic-exporter	1.1.2
logs	2.0.7
kibana	1.5.2
grafana	2.2.0
prometheus	8.8.0
prometheus-node-exporter	1.4.0
metallb	0.8.4
metrics-server	2.4.0

Software	Chart Version
jaeger	0.13.3

Note:

If any of the software specified in the above table is not installed in CNE, install the software.
If OCCNE is the platform, then refer *Cloud Native Environment Installation Guide*.

Network Access

The Kubernetes cluster host must have network access to:

- Local docker image repository where the NSSF images are available.
- Local helm repository where the NSSF helm charts are available.

Note:

All the kubectl and helm related commands that are used in this document must be executed on a system depending on the infrastructure or the deployment. It could be a client machine.

Machine Client Requirements

Following are the requirements for the client machine where the deployment commands need to be executed:

- It must have network access to the helm repository and docker image repository.
- Helm repository must be configured on the client.
- It must have network access to the Kubernetes cluster.
- It must have necessary environment settings to run the `kubectl` commands. The environment must have privileges to create namespace in the Kubernetes cluster.
- It must have helm client installed. The environment must be configured so that the `helm install` command deploys the software in the Kubernetes cluster.

NSSF Pre-deployment Configuration

This section describes about the various steps of pre-deployment configuration for NSSF. It includes the following:

1. Verify and Create kubernetes Namespace
2. Create MySql Database and User for OCNSSF
3. Create a Kubernetes Secret for Storing Database Username and Password
4. Create Private Keys and Certificates for Ingress Gateway and Egress Gateway

5. Create Private Key and Certificates to enable OAuth

Verify and Create Kubernetes Namespace

This section explains how user can verify required namespace exists in system or not. If namespace does not exist, user must create it.

Procedure

1. Verify whether required namespace already exists in system by executing the following command:

   ```
    $ kubectl get namespaces
   ```

2. If the output of the above command does not display the required namespace then create the namespace by executing following command:

   ```
   $ kubectl create namespace <required namespace>
   ```

   Example:

   ```
   $ kubectl create namespace ocnssf
   ```

Create MySQL Database and User for OCNSSF

1. Login to the server or machine which has permission to access the SQL nodes of NDB cluster.
2. Connect to the SQL nodes of NDB cluster one by one.
3. Login to the MySQL prompt using root permission or user, who has permission to create users with permissions as mentioned below. Example: `mysql -h 127.0.0.1 -uroot -p`
4. Check whether OCNSSF network function user already exists. If not exists, create an OCNSSF network function user by executing following queries:
   a. Execute `$ SELECT User FROM mysql.user;` to list the users.
   b. If user does not exist, create the new user by executing `$ CREATE USER '<OCNSSF User Name>'@'%' IDENTIFIED BY '<OCNSSF Password>';` Example: `$ CREATE USER 'nssfusr'@'%' IDENTIFIED BY 'nssfpasswd';`
5. Check OCNSSF network function database already exists. If does not exist, create an OCNSSF network function database and provide permissions to OCNSSF user name created in the previous step:
   a. Execute `$ show databases;` to check if database exists.
   b. If database does not exists, execute `$ CREATE DATABASE IF NOT EXISTS <OCNSSF Database> CHARACTER SET utf8;` for Database creation. Example:

      ```
      $ CREATE DATABASE IF NOT EXISTS nssfdb CHARACTER SET utf8;
      ```

   c. Granting permission to user:

      ```
      $ GRANT SELECT,INSERT,CREATE,ALTER,DROP,LOCK TABLES,CREATE
      TEMPORARY TABLES, DELETE,UPDATE,EXECUTE ON <OCNSSF Database>.* TO
      '<OCNSSF User Name>'@'%';
      ```

Create a Kubernetes Secret for Storing Database Username and Password

1. Create a yaml file with the username and password with the syntax shown below:

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: ocnssf-db-creds
   type: Opaque
   data:
     mysql-username: bnNzZnVzcg==
     mysql-password: bnNzZnBhc3N3ZA==
     mysql-db-name: bnNzZmRi
   ```

   > **Note:**
   >
   > The values for **mysql-username** and **mysql-password** must be **base64** encoded.

2. Execute the following command to create the kubernetes secret:

   kubectl create -f yaml_file_name -n namespace

   where:

   yaml_file_name is a name of the yaml file that is created in step 1.

   namespace is the deployment namespace used by the helm command.

3. Execute the following command to verify the secret creation:
   $ kubectl describe secret <database secret name> -n <Namespace of MYSQL secret>

   Example:*$ kubectl describe secret ocnssf-db-creds -n ocnssf*

Create Private Keys and Certificates for Ingress Gateway and Egress Gateway

This section describes how to create private keys and certificates in NSSF.

Creating Private Key and Certificates to enable https

To create private keys and certificates:

1. Generate RSA private key by executing the following command:

   ```
   openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout
   rsa_private_key -out rsa_certificate.crt
   ```

2. Convert private key to .pem format by executing the following command:

   ```
   openssl rsa -in rsa_private_key -outform PEM -out rsa_private_key_pkcs1.pem
   ```

3. Generate certificate using the private key by executing the following command:

```
openssl req -new -key rsa_private_key -out ocegress.csr -config ssl.conf
```

> **Note:**
>
> The `ssl.conf` can be used to configure default entries along with storage area network (SAN) details for your certificate.

A sample of the `ssl.conf` is provided below:

```
#ssl.conf
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
req_extensions = req_ext

[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = IN
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Karnataka
localityName = Locality Name (eg, city)
localityName_default = Bangalore
organizationName = Organization Name (eg, company)
organizationName_default = Oracle
commonName = Common Name (e.g. server FQDN or YOUR name)
commonName_max = 64
commonName_default = localhost

[ req_ext ]
subjectAltName = @alt_names

[alt_names]
IP = 127.0.0.1
DNS.1 = localhost
```

4. Create root certificate authority (CA) by executing the following set of commands:

```
openssl req -new -keyout cakey.pem -out careq.pem
openssl x509 -signkey cakey.pem -req -days 3650 -in careq.pem -out
caroot.cer
-extensions v3_ca
echo 1234 > serial.txt
```

5. Sign the server certificate with root CA private key by executing the following command:

```
openssl x509 -CA caroot.cer -CAkey cakey.pem -CAserial serial.txt -req -
in
ocegress.csr -out ocegress.cer -days 365 -extfile ssl.conf -extensions
req_ext
```

> **Note:**
>
> The **ssl.conf** file must be reused, as SAN contents is not packaged when signing.
>
> 1. Create **key.txt** by entering any password.
> 2. Create **trust.txt** by entering any password.

Creating a Secret

Note: User must create a secret for database access before deploying NSSF.

To create a secret:

1. Execute `kubectl get namespace` to list the namespaces.
2. If name space does not exist, create a new namespace by executing the following command:

   `kubectl create namespace <NameSpace>`
   where:

   namespace is the deployment namespace used by the helm command.
3. Generate secret out of the keys and certificates by executing the following command:

   ```
   kubectl create secret generic k8SecretName --from-
   file=rsa_private_key_pkcs1.pem --from-file=trust.txt --
   from-file=key.txt --from-file=ocegress.cer
   --from-file=caroot.cer -n k8NameSpace
   ```

   where:

   k8NameSpace is the deployment namespace used by the helm command.

   k8SecretName is the name of secret generated.

   Example:

   ```
   kubectl create secret generic accesstoken-secret --from-
   file=rsa_private_key_pkcs1.pem --from-file=trust.txt --from-
   file=key.txt
   --from-file=ocegress.cer --from-file=caroot.cer -n ocnssf
   ```

Create Private Key and Certificates to Enable OAuth

1. Auth token generator, OCNRF provides its public key to NSSF.
2. User must create kubernetes secret to store NRF public key.
   There can be multiple NRF public keys.

   File Name or key (this key is of map) of Public Key must be in the following format:

   `"{nrfInstanceId}_{SigningAlgorithm}.pem"`

   where `nrfInstanceId`is Instance Id of NRF.

`SigningAlgorithm` can have following values:

```
ES256: ECDSA using P-256 and SHA-256
ES384: ECDSA using P-384 and SHA-384
ES512: ECDSA using P-521 and SHA-512
RS256: RSASSA-PKCS-v1_5 using SHA-256
RS384: RSASSA-PKCS-v1_5 using SHA-384
RS512: RSASSA-PKCS-v1_5 using SHA-512
PS256: RSASSA-PSS using SHA-256 and MGF1 with SHA-256
PS384: RSASSA-PSS using SHA-384 and MGF1 with SHA-384
PS512: RSASSA-PSS using SHA-512 and MGF1 with SHA-512
```

3. Generate secret out of the keys and certificates by executing the following command:

   ```
   kubectl create secret generic <Secret_Name> --from-file={nrfInstanceId}_{SigningAlgorithm}.pem-n <NameSpace>
   ```

   Example:

   ```
   kubectl create secret generic nrfpublickeysecret --from-file=fe7d992b-0541-4c7d-ab84-c6d70b1b01b1_RS256.pem-n ocnssf
   ```

NSSF Installation Sequence

This section provides details about the sequence in which NSSF must be installed.

Table 4-1 Installation Sequence

Sl.No	Phase	Description
1	Installation Preparation	Download the required files and load the files to the system.
2	Configure `custom_values.yaml` file.	This includes configuring the following based on the deployment: 1. Repository path 2. Primary and Secondary node 3. NSSF details **Note:** Other configurations might be changed based on the deployment.
3	NSSF deployment	NSSF can be deployed in either of the following ways: • With HELM repository • With HELM tar
4	Verify NSSF deployment	Check if the services and pods are up and running.

Docker Images for NSSF

Following are the NSSF images:

Table 4-2 NSSF Images

Services	Docker Image Name
Egress Gateway	ocnssf-egress
Ingress Gateway	ocnssf-ingress
NS-Availability	ocnssf-nsavailability
NS-Config	ocnssf-nsconfig
NS-Selection	ocnssf-nsselection
NS-Subscription	ocnssf-nssubscription
NRF Client Service	ocnssf-nrf-clientservice
Application Info Service	ocnssf-appinfo
Configuration Server Service	config_server
Performance Monitoring Service	perf_info

NSSF Installation Preparation

The following procedure describes the steps to download the NSSF Images and Helm files from OSDC.

For more information about configuring docker image and registry, refer chapter OCCNE Docker Image Registry Configuration in OCCNE Installation Guide.

Table 4-3 NSSF Installation Preparation

Step	Procedure	Description

1	Download the NSSF package file	Customers are required to download the NSSF package file from Oracle Software Delivery Cloud (OSDC). Package is named as follows:
		`<nfname>-pkg-<marketing-release-number>`
		`Example: ocnssf-pkg-1.3.0.0.0.tgz`

Table 4-3 (Cont.) NSSF Installation Preparation

Step	Procedure	Description
2	Untar the NSSF Package File	Untar the NSSF package to the specific repository: `tar -xvf <<nfname>-pkg-<marketing-release-number>>` The package file consists of following: 1. NSSF Docker Images File `ocnssf-images-1.3.0.tar` 2. Helm File `ocnssf-1.3.0.tgz` 3. Readme txt file `Readme.txt` (Contains cksum and md5sum of tarballs)
3	Check the checksums	Check the checksums of tarballs mentioned in `Readme.txt`. Refer to the `Readme.txt` file for commands and checksum details.
4	Load the tarball to system	Execute the following command to push the Docker images to docker registry: `docker load --input ocnssf-images-1.3.0.tar`
5	Check if all the images are loaded	Execute the following command to check: `docker images` Refer table **NSSF Images** in section NSSF Installation Sequence for the list of images.

6	Push docker images to docker registry	Execute the following commands to push the docker images to docker registry:
		`docker tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>`
		`docker push <docker-repo>/<image-name>:<image-tag>`

7	Untar Helm Files	Untar the helm files: `tar -xvzf ocnssf-1.3.0.tgz`

Table 4-3 (Cont.) NSSF Installation Preparation

Step	Procedure	Description
8	Download the Native Network Slice Selection Function (NSSF) Custom Template ZIP file	Download the **Native Network Slice Selection Function (NSSF) Custom Template** ZIP file from OHC: • Go to the URL, docs.oracle.com • Navigate to **Industries >Communications >Signaling & Policy >Cloud Native Core** • Click the **Native Network Slice Selection Function (NSSF) Custom Template** link to download the zip file. • Unzip the template to get the following files: – `ocnssf-custom-values-1.3.0.yaml` – `onssfDashboard.json`

NSSF Installation

This section includes information about NSSF deployment.

Following are the parameters and definitions used during NSSF deployment:

Table 4-4 Parameters and Definitions

Parameters	Definitions
`<helm chart>`	It is the name of the chart that is of the form `<helm repo>/ocnssf`.
`<OCNSSF version>`	It is the software version (helm chart version) of the NSSF. This is optional, if omitted, the default version is the latest version available in helm repository.
`<release>`	It is a name provided by the user to identify the helm deployment.
`<k8s namespace>`	It is a name provided by the user to identify the kubernetes namespace of the NSSF. All the NSSF micro services are deployed in this kubernetes namespace.
`<mysql host>`	It is the hostname of the `mysql` service and can be provided as, `<release>-mysql.<k8s namespace>`.

NSSF Deployment on Kubernetes

Note:

- To configure the parameters, refer Customizing NSSF .

Create Database User/Group

The NSSF uses a MySQL database to store the configuration and run time data.

The NSSF deployment using MySQL NDB cluster requires the database administrator to create a user in the MYSQL DB and to provide the user with necessary permissions to access the tables in the NDB cluster.

1. Login to the server where the ssh keys are stored and the SQL nodes are accessible.
2. Connect to the SQL nodes.
3. Login to the Database as a root user.
4. Create a user and assign it to a group having necessary permission to access the tables on primary SQL nodes:

   ```
   CREATE USER '<username>'@'%' IDENTIFIED BY '<password>';
   DROP DATABASE if exists nssfdb;
   CREATE DATABASE nssfdb CHARACTER SET utf8;
   GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, CREATE TEMPORARY
   TABLES, DELETE, UPDATE,
   EXECUTE ON nssfdb.* TO '<username>'@'%';
   USE nssfdb;
   ```

5. Grand necessary permissoins to access the tables on seconary SQL nodes:

   ```
   GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, CREATE TEMPORARY
   TABLES, DELETE, UPDATE,
   EXECUTE ON nssfdb.* TO '<username>'@'%';
   USE nssfdb;
   ```

   > **Note:**
   >
   > The `<username>` and `<password>` is created by the Database Administrator.

6. Exit from database and logout from SQL node.

Table 4-5 NSSF Deployment

Step #	Procedure	Description
1	Create customized `ocnssf-custom-values-1.3.0.yaml` file	Create the customized `ocnssf-custom-values-1.3.0.yaml` with the required input parameters. To configure the `ocnssf-custom-values-1.3.0.yaml`, refer Customizing NSSF or, The `ocnssf-custom-values-1.3.0.yaml` template can be downloaded from OHC. Download the package `ocnssf-custom-configTemplates-1.3.0.0.0.zip` and Unzip to get `ocnssf-custom-values-1.3.0.yaml` file.
2	Go to the unzipped OCNSSF package	Go to the following directory: `cd OCNSSF-pkg-1.3.0.0.0`

3	Deploy OCNSSF	Execute the following command:
		`helm install ocnssf/ --name <helm-release> --namespace <k8s namespace> -f <ocnssf_customized_values.yaml>`
		Example:
		`helm install ocnssf/ --name ocnssf --namespace ocnssf -f ocnssf-custom-values-1.3.0.yaml`

4	Check status of the deployment	`helm status --name <helm-release>` Example: `helm status --name ocnssf`

Table 4-5 (Cont.) NSSF Deployment

<table>
<tr><th>Step #</th><th>Procedure</th><th>Description</th></tr>
<tr><td>5</td><td>Check status of the services</td><td>

Execute the following command:

```
kubectl -n <k8s namespace> get services
```

Example:

```
kubectl -n ocnssf get services
```

Note: If `metallb` is used, `EXTERNAL-IP` is assigned to<helm release name>-endpoint. `ocnssf` is the helm release name.

```
NAME            TYPE             CLUSTER-IP    EXTERNAL-IP
PORT(S)                    AGE
ocnssf-appinfo          ClusterIP    10.103.32.93
<none>    5906/TCP                     4h22m
ocnssf-config-server    ClusterIP    10.103.227.15
<none>    5807/TCP,9000/TCP            4h22m
ocnssf-egress           ClusterIP    10.107.83.168
<none>    8080/TCP,5701/TCP            4h22m
ocnssf-ingress          LoadBalancer 10.98.78.95
<pending> 80:30075/TCP,5701:32531/TCP  4h22m
ocnssf-nrf-clientserviceNodePort     10.99.244.169
<none>    5910:30025/TCP,5805:31152/TCP4h22m
ocnssf-nsavailability   NodePort     10.98.166.211
<none>    5745:30592/TCP               4h22m
ocnssf-nsconfig         NodePort     10.106.24.192
<none>    5755:32160/TCP               4h22m
ocnssf-nsdb             ClusterIP    10.107.148.15
<none>    3306/TCP                     4h22m
ocnssf-nsselection      NodePort     10.107.230.118
<none>    5745:31263/TCP,4546:31199/TCP4h22m
ocnssf-nssubscription   NodePort     10.100.116.162
<none>    5745:31674/TCP,4546:31542/TCP4h22m
ocnssf-performance      NodePort     10.98.216.252
<none>    5905:31402/TCP
```

</td></tr>
</table>

Table 4-5 (Cont.) NSSF Deployment

Step #	Procedure	Description

6	Check status of the pods	Execute the following command:
		`kubectl get pods -n <k8s namespace>`
		Status column of all the pods must indicate 'Running'.
		Ready column of all the pods should be n/n, where n is number of containers in the pod.
		Example:
		`kubectl get pods -n ocnssf`
		
		`NAME                            READY   STATUS RESTARTS AGE`
		`ocnssf-appinfo-7969c9fbf7-4fmgj          1/1  Running`
		`0   18m`
		`ocnssf-config-server-54bf4bc8f9-s82cv    1/1  Running`
		`0   18m`
		`ocnssf-egress-6b6bff8949-2mf7b           1/1  Running`
		`0   18m`
		`ocnssf-ingress-68d76954f5-9fsfq          1/1  Running`
		`0   18m`
		`ocnssf-nrf-clientservice-7c5fcc9588-ckmmr1/1  Running`
		`0   18m`
		`ocnssf-nsavailability-644999bbfb-9gcm5   1/1  Running`
		`0   18m`
		`ocnssf-nsconfig-577446c487-dzsh6         1/1  Running`
		`0   18m`
		`ocnssf-nsdb-585f7bd7d-tdth4              1/1  Running`
		`0   18m`
		`ocnssf-nsselection-5dfcc94bc7-q9gct      1/1  Running`
		`0   18m`
		`ocnssf-nssubscription-5c898fbbb9-fqcw6   1/1  Running`
		`0   18m`
		`ocnssf-performance-6d75c7f966-qm5fq      1/1  Running`
		`0   18m`

5
Customizing NSSF

The OCNSSF deployment can be customized by overriding the default values of various configurable parameters.

A `ocnssf_values.yaml` file can be prepared to customize the parameters. The section **NSSF Configurable Parameters** is an example of OCNSSF customization file.

Configuration Options During Deployment

Basic Configuration:

1. Once docker platform configurations are done, proceed as per NSSF Configurable Parameters .
2. Check Registry is in place and contains latest helm charts and jar as per the release for NSSF node.

NSSF Configurable Parameters

NS-Selection

Table 5-1 NS-Selection

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
omeMcc	MCC of PLMN of Home network		M	3 digit integer value	Used when Ns-Selection GET request comes without TAI
homeMnc	MNC of PLMN of Home network		M	2/3 digit integer value	Used when Ns-Selection GET request comes without TAI
nrfUrl	URL of NRF		M	Valid URL	
reqnftime	When set to true AMF can send current time as Http Header	FALSE	O	TRUE/ FALSE	This field is used when time based network slice is enabled. If set to true time sent by AMF is used to get time profile based slice When not then current local time of NSSF is used to get Slice.

Table 5-1 (Cont.) NS-Selection

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
outboundProxy	Value of outbound proxy for NSSF		O	Host-name/IP address:port of outbound proxy	
features.nrfdiscovery	Flag to enable / disable NRF discovery for each GET request on NS-Selection Initial Register and Update Config request	FALSE	O	TRUE/ FALSE	
features.relevance	Flag to enable / disable Relevance feature	FALSE	O	TRUE/ FALSE	When enabled, in conjection with features.candidateResolution. NSSF will apply relevance algorithm to select/sort Candidate AMFs as a response to Initial register or UE config update request which are part of selected Target AMF Set.
features.candidateResolution	Flag to enable / disable Candidate Resolution feature	FALSE	O	TRUE/ FALSE	When this feature is set to false NSSF returns TargetAMFSetId and TargetAMFRegionId for NS-Selection GET request for Initial Register message and UE-Config update. When this feature is set to true NSSF computes and returns Candidate AMF list for NS-Selection GET request for Initial Register message and UE-Config update.
nrfDiscoveryProperties.disclimit	Max Number of AMFs set on NRF discovery request	5	Mandatory when features.nrfdiscovery is set to true	2-10	This is accepted only when nrfDiscovery is set to true.

Table 5-1 (Cont.) NS-Selection

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
candidateResolutionProperties.maxcandidates:	Maximum number of candidate AMFs	3	Mandatory when features.candidateResolution is set to true	2-10	This value is accepted only when candidateResolution is enabled.
global.databaseSecretName	This parameter is the name of Kubectl secret which contains Username and password for Database.		M	Kubernetes Secret file name	Creation of Secrets must be done before installation of NSSF.
mysql.primary.host	Primary MYSQL Host IP or Hostname	ocnssf-mysq	M	Primary Mysql HostName or IP	OCNSSF will connect Primary MYSQL if not available then it will connect secondary host. For MYSQL Cluster use respective IP Address or Mysql Host or Service
mysql.secondary.host	Secondary MYSQL Host IP or Hostname	ocnssf-mysql	M	Secondary Mysql HostName or IP	For MYSQL Cluster use respective Secondary IP Address or Mysql Host or Service
mysql.port	Port of MYSQL Database	3306	M	Port of MySQL Database	
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

NS-Availability

Table 5-2 NS-Availability

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
maxExpiryDuration	Max duration (in Hours) upto which AMF can subscribe to NSSF	240	O	100-1000	Max Expiry duration must be more than Min Expiry duration. Requesting more than max expiry duration will be gruanted the value which is configured.
minExpiryDuration	Min duration (in Hours) of a valid subscription towards NSSF	0	O	0-100	Request lesser than configured value shall be rejected.
global.databaseSecretName	This parameter is the name of Kubectl secret which contains Username and password for Database.		M	Kubernetes Secret file name	Creation of Secrets must be done before installation of NSSF.
mysql.primary.host	Primary MYSQL Host IP or Hostname	ocnssf-mysq	M	Primary Mysql HostName or IP	OCNSSF will connect Primary MYSQL if not available then it will connect secondary host. For MYSQL Cluster, use respective IP Address or Mysql Host or Service.
mysql.secondary.host	Secondary MYSQL Host IP or Hostname	ocnssf-mysql	M	Secondary Mysql HostName or IP	For MYSQL Cluster, use respective Secondary IP Address or Mysql Host or Service.
mysql.port	Port of MYSQL Database	3306	M	Port of MySQL Database	
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

NS-Config

Table 5-3 NS-Config

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
nrf: subscription	Flag to enable subscription to NRF based on Target AMF set and Region Id	TRUE	M	TRUE/ FALSE	When set to true, NSSF subscribes to get all the AMFs added/deleted on Target AMF set and Target AMF region is configured to NRF. NS-Policy: nrfDiscovery and NS-Config: nrf: Subscription are mutually exclusive.
notificationHandlerUrl	URL at which NS-Config MS receives notifications		When nrf.subscription is set to true then Mandatory	Valid URL	This is the URL where NRF sends notifications when nrf:subscription is set to true.
mysql.primary.host	Primary MYSQL Host IP or Hostname	ocnssf-mysql	M	Primary Mysql HostName or IP	OCNSSF will connect Primary MYSQL if not available then it will connect secondary host. For MYSQL Cluster use respective IP Address or Mysql Host or Service.
global.databaseSecretName	This parameter is the name of Kubectl secret which contains Username and password for Database.		M	Kubernetes Secret file name	Creation of Secrets must be done before installation of NSSF.
mysql.secondary.host	Secondary MYSQL Host IP or Hostname	ocnssf-mysql	M	Secondary Mysql HostName or IP	For MYSQL Cluster use respective Secondary IP Address or Mysql Host or Service.
mysql.port	Port of MYSQL Database	3306	M	Port of MySQL Database	
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

NS-Subscription

Table 5-4 NS-Subscription

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Note
httpMaxRetries	Number of retry s to be done when AMF does not respond to Notification.	3	M	2-5	
global.databaseSecretName	This parameter is the name of Kubectl secret which contains Username and password for Database.		M	Kubernetes Secret file name	Creation of Secrets must be done before installation of NSSF.
mysql.primary.host	Primary MYSQL Host IP or Hostname	ocnssf-mysq	M	Primary Mysql HostName or IP	OCNSSF connects Primary MYSQL, if not available then it will connect secondary host. For MYSQL Cluster use respective IP Address or Mysql Host or Service
mysql.secondary.host	Secondary MYSQL Host IP or Hostname	ocnssf-mysql	M	Secondary Mysql HostName or IP	For MYSQL Cluster use respective Secondary IP Address or Mysql Host or Service

Table 5-4 (Cont.) NS-Subscription

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Note
mysql.port	Port of MYSQL Database	3306	M	Port of MySQL Database	
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

Common Micro Services

Ingress Gateway

Table 5-5 Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
global.dockerRegistry	Name of the Docker registry which hosts Ingress docker images.	NA	M		This is the registry which has docker images. Change this value if there is a need.
global.type	type of service	LoadBalancer	M	ClusterIP, NodePort, LoadBalancer and ExternalName	
global.serviceAccountName	Service Account name	''	O		
global.metalLbIpAllocationEnabled	Enable or disable IP Address allocation from Metallb Pool	true	O		

Table 5-5 (Cont.) Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
global.metalLbIpAllocationAnnotation	Address Pool Annotation for Metallb	metallb.universe.tf/ address-pool: signaling	No		
global.staticIpAddressEnabled	If Static load balancer IP needs to be set, then set staticIpAddressEnabled flag to true and provide value for staticIpAddress Else random IP will be assigned by the metalLB from its IP Pool	false	No		
global.staticIpAddress	StaticIp	10.75.212.60			
global.publicHttpSignalingPort	Http Signalling port	80	M		
global.publicHttpsSignallingPort	Https Signalling port	443	M		
global.staticNodePortEnabled	Node Port Enabled	true	No		
global.staticHttpNodePort	Http Node Port	30075	M		
global.staticHttpsNodePort	Https Node Port	30043	M		
enableOutgoingHttps	Enabling it for outgoing https request	false	M		Change it to true for enabling https for outgoing requests.
enableIncomingHttp	Enabling it for incoming http request	false	M		
enableIncomingHttps	Enabling it for incoming https request	true	M		
enablehttp1	Enable it for http1.1	false	No		Change it to true to enable

Table 5-5 (Cont.) Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
dnsRefresh Delay	Dns Refresh Delay in milli-seconds	120000	No		
oauthValidatorEnabled	Oauth Validator Enabled	false	M		Change it to true to enable oauth
jaegerTracingEnabled	Enable jaeger tracing	false	No		Change it to true if needed.
openTracing.jaeger.udpSender.host	Jaeger Host	jaeger-agent.cne-infra	M (If jaegerTracingEnabled is true)		
openTracing.jaeger.udpSender.port	Jaeger Port	6831	M (If jaegerTracingEnabled is true)		
openTracing.jaeger.probabilisticSampler	Jaeger sampling frequency	0.5	M (If jaegerTracingEnabled is true)		
nfType	NFType of service producer.	Value to be updated accordingly	M (When oauthValidatorEnabled)		
nfInstanceId:	NF InstanceId of service producer.	Value to be updated accordingly	M (When oauthValidatorEnabled)		
producerScope:	Comma-seperated list of services hosted by service producer.	Value to be updated accordingly	M (When oauthValidatorEnabled)		
allowedClockSkewSeconds	set this value if clock on the parsing NF(producer) is not perfectly in sync with the clock on the NF(consumer) that created the JWT.	0	M (When oauthValidatorEnabled)		
nrfPublicKeyKubeSecret	Name of the secret which stores the public key(s) of NRF.	Value to be updated accordingly	M (When oauthValidatorEnabled)		
nrfPublicKeyKubeNamespace	Namespace of the NRF publicKey Secret	Value to be updated accordingly	M (When oauthValidatorEnabled)		

Table 5-5 (Cont.) Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
validationType	Values can be "strict" or "relaxed"."strict" means that incoming request without "Authorization"(Access Token) header will be rejected. "relaxed" means that if incoming request contains "Authorization" header, it will be validated. If incoming request doesnot contain"Authorization" header, validation will be ignored.	Value to be updated accordingly	M (When oauthValidatorEnabled)		
producerPlmnMNC	MNC of service producer.	Value to be updated accordingly	No		
producerPlmnMCC	MCC of service producer.	Value to be updated accordingly	No		
cnccIamEnabled	CNCC Identity-Access-Management(IAM)	false	No		Change it to true if required
ingressGwCertReloadEnabled		true	No		
rateLimiting.enabled	Ratelimiting feature enabled	true	No		
routeRateLimiting.enabled	Route based ratelimiting feature enabled	true	No		
globalIngressRateLimiting.enabled	Global rate limiting is enabled	true	No		
globalIngressRateLimiting.duration	Iterations of time duration(In seconds) for which bucketCapacity and refillRate are reset.	1(in seconds)	M(if globalIngressRateLimiting.enabled)		

Table 5-5 (Cont.) Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
globalIngressRateLimiting.burstCapacity	Holds maximum number of tokens in the bucket for the given duration.	1	M(if globalIngressRateLimiting.enabled)		
globalIngressRateLimiting.refillRate	Number of tokens to be added to the bucket for the given duration	1	M (if globalIngressRateLimiting.enabled)		
ssl.privateKey.k8SecretName	Name of the privatekey secret	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.privateKey.k8NameSpace	Namespace of privatekey	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.privateKey.rsa.fileName	rsa private key file name	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.privateKey.ecdsa.fileName	ecdsa private key file name	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.certificate.k8SecretName	Name of the privatekey secret	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.certificate.k8NameSpace	Namespace of privatekey	n/a	M (If enableIncomingHttps is true otherwise No)		

Table 5-5 (Cont.) Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
ssl.certificate.rsa.fileName	rsa private key file name	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.certificate.ecdsa.fileName	ecdsa private key file name	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.caBundle.k8SecretName	Name of the privatekey secret	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.caBundle.k8NameSpace	Namespace of privatekey	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.caBundle.rsa.fileName	rsa private key file name	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.keyStorePassword.k8SecretName	Name of the privatekey secret	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.keyStorePassword.k8NameSpace	Namespace of privatekey	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.keyStorePassword.fileName	File name that has password for keyStore	n/a	M (If enableIncomingHttps is true otherwise No)		

Table 5-5 (Cont.) Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
ssl.trustStorePassword.k8SecretName	Name of the privatekey secret	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.trustStorePassword.k8NameSpace	Namespace of privatekey	n/a	M (If enableIncomingHttps is true otherwise No)		
ssl.trustStorePassword.fileName	File name that has password for trustStore	n/a	M (If enableIncomingHttps is true otherwise No)		
id	id of the route		M		
uri	Service name of the internal microservice of this NF		M		
path	Provide the path to be matched.		M		
order	Provide the order of the execution of this route.		M		
methodRateLimiting.method[0]	Method on which ratelimiting is applicable		M (if routeRateLimiting.enabled)		
methodRateLimiting.burstCapacity[0]	burstCapacity		M (if routeRateLimiting.enabled)		
methodRateLimiting.refillRate[0]	Refill rate		M (if routeRateLimiting.enabled)		
methodRateLimiting.duration[0]	Duration		M (if routeRateLimiting.enabled)		
image.repository	Full Image Path		M	Full image path of image	

Table 5-5 (Cont.) Ingress Gateway

Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

Egress gateway

Table 5-6 Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
openTracing.jaeger.probabilisticSampler		0.5	M (If jaegerTracingEnabled is true)		
maxConnectionsPerIp	Max Connections allowed per Ip	4	No		
connectionTimeout	Connection timeout in milli seconds	1000	No		
maxConnectionsQueuedPerDestination	jetty client configuration	1024	No		
openTracing.jaeger.udpSender.port	Jaeger Port	6831	M (If jaegerTracingEnabled is true)		
serviceEgressGateway.port		8080	No		
serviceEgressGateway.sslPort	SSL Port	8442	No		
serviceEgressGateway.actuatorPort	Actuator Port	9090	No		
global.serviceAccountName	Service Account Name	''	No		

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
cipherSuites	Supported Cipher Suites in Egress	- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	No		Connection with other ciphers would be rejected.

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
scp.instances.http[0].apiPrefix	First Scp instance apiPrefix. Change this value to corresponding prefix if "/" is not expected to be provided along. Applicable only for SCP with TLS enabled.	/	No		Examples : XXX, Point to be noted here is that / is not required to be included when providing some data.
scp.instances.https[0].apiPrefix	First Scp instance apiPrefix. Change this value to corresponding prefix if "/" is not expected to be provided along. Applicable only for SCP with TLS enabled.	/	No		Examples : XXX, Point to be noted here is that / is not required to be included when providing some data.
type	Type of service	ClusterIP	M		Possible values are :- ClusterIP, NodePort, LoadBalancer and ExternalName
enableOutgoingHttps	Enabling it for outgoing https request	false	No		Change it to true for enabling https for outgoing requests.
K8ServiceCheck	Enable this if loadbalancing is to be done by egress instead of K8s	false	No		
headlessServiceEnabled	Enabling this will make the service type default to ClusterIP	false	No		
jaegerTracingEnabled	Enable jaeger tracing	false	No		Change it to true if needed.
notificationRateLimit.enabled	Flag to enable rate limiting for "notification" type of messages.	false	No		

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
globalretry.enabled	Can be set to true if Scp re-route feature(scpRerouteEnabled) is enabled.	false	No		
scp.scpDefaultScheme	Default scheme applicable when 3gpp-sbi-target-apiroot header is missing	https	No		
openTracing.jaeger.udpSender.host	Jaeger Host	jaeger-agent.cne-infra	M (If jaegerTracingEnabled is true)		
nfType	NFType of service consumer.	Modify the field with actual value , required if oAuth is enabled.	M		
consumerPlmnMNC	MNC of service Consumer.	Modify the field with actual value , required if oAuth is enabled.	No		
consumerPlmnMCC	MCC of service Consumer.	Modify the field with actual value , required if oAuth is enabled.	No		

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
nrfAuthority	NRF's ${HOSTNAME}:{PORT}	Modify the field with actual value, required if oAuth is enabled.	M		
nfInstanceId:	NF InstanceId of Service Consumer.	Modify the field with actual value, required if oAuth is enabled.	M		
ssl.privateKey.k8SecretName	Name of the privatekey secret	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.privateKey.k8NameSpace	Namespace of privatekey	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.privateKey.rsa.fileName	rsa private key file name	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.privateKey.ecdsa.fileName	ecdsa private key file name	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.certificate.k8SecretName	Name of the privatekey secret	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.certificate.k8NameSpace	Namespace of privatekey	n/a	M (If enableOutgoingHttps is true otherwise No)		

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
ssl.certificate.rsa.fileName	rsa private key file name	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.certificate.ecdsa.fileName	ecdsa private key file name	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.caBundle.k8SecretName	Name of the privatekey secret	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.caBundle.k8NameSpace	Namespace of privatekey	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.caBundle.rsa.fileName	rsa private key file name	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.keyStorePassword.k8SecretName	Name of the privatekey secret	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.keyStorePassword.k8NameSpace	Namespace of privatekey	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.keyStorePassword.fileName	File name that has password for keyStore	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.trustStorePassword.k8SecretName	Name of the privatekey secret	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.trustStorePassword.k8NameSpace	Namespace of privatekey	n/a	M (If enableOutgoingHttps is true otherwise No)		
ssl.trustStorePassword.fileName	File name that has password for trustStore	n/a	M (If enableOutgoingHttps is true otherwise No)		

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
scp.instances.http[0].host	First Scp instance HTTP IP/FQDN	NA	M(If "scp.scpIntegrationEnabled" is set to true.)		More SCP instances can be configured in a similar way if required.
scp.instances.http[0].port	First Scp instance Port	NA	M(If "scp.scpIntegrationEnabled" is set to true.)		
scp.instances.https[0].host	First Scp instance HTTPS IP/FQDN	NA	M(if "scp.scpIntegrationEnabled" is set to true.)		More SCP instances can be configured in a similar way if required.
scp.instances.https[0].port	First Scp instance HTTPS Port	NA	M(if "scp.scpIntegrationEnabled" is set to true.)		
global.dockerRegistry	Name of the Docker registry which hosts Egress docker images.	ocnrf-registry.us.oracle.com:5000	M		Ideally this is the registry which has docker images. Change this value if there is a need.
global.appinfoServiceEnable	Enabled to get RBAC permission for k8s apiserver communication	true	M		
scp.scpIntegrationEnabled	Change this to false when scp integration is not required	true	No		
scp.scpRerouteEnabled	Set this flag to true if re-routing to multiple SCP instances is to be enabled.	true	No		
oauthClientEnabled	Flag to enable or disable oauth client. If not modified, Default value 'false' will be defaulted.	true	No		Change it to true to enable Oauth
egressGwCertReloadEnabled		true	No		

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
notificationRateLimit.duration	Iterations of time duration(In seconds) for which bucketCapacity and refillRate are reset.		M(If notificationRateLimit.enabled is set to true)		
notificationRateLimit.bucketCapacity	Holds maximum number of tokens in the bucket for the given duration.		M(If notificationRateLimit.enabled is set to true)		
notificationRateLimit.refillRate	Number of tokens to be added to the bucket for the given duration		M(If notificationRateLimit.enabled is set to true)		
globalretry.retries	Number of re-routes to be attempted to alternate SCP instances and this property will be considered in the absence of "routesConfig[0].filterName2.retries" attribute at route level.		M(If "routesConfig[0].filterName2.retries" is not defined)		
routesConfig[0].id	Id of the route		M		Can be any name of your choice. Note: Multiple routes can be configured in a similar way.
routesConfig[0].uri	Provide any dummy url , existing url can also left with existing value		M		Please note provided sample url does not make any impact (http or https) as url's will be constructed in the code.
routesConfig[0].path	Provide the path to be matched.		M		
routesConfig[0].order	Provide the order of the execution of this route.		M		

Table 5-6 (Cont.) Egress gateway

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
routesConfig[0].filterName1	Provide filtername as "ScpFilter"		M (If scpintegrationenabled is true)		If FilterName1 is not provided then it would be considered as direct Egress Gateway path and configured accordingly during deployment.
routesConfig[0].filterName2.name	Provide filtername as "ScpRetry"		M (If scpRerouteEnabled is true)		With out FilterName1 , it is not possible to configure FilterName2.name
routesConfig[0].filterName2.retries	Number of re-routes to be attempted to alternate SCP instances if request matches this route's path.		M (If scpRerouteEnabled is true)		If this is not defined then globalretry.retries parameter is applicable when globalretry.enabled is true.
routesConfig[0].filterName2.methods	The type of methods for which the re-route need to be attempted.		M (If scpRerouteEnabled is true)		
routesConfig[0].filterName2.statuses	The type response error codes on which the re-route need to be attempted.		M (If scpRerouteEnabled is true)		
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

Nrfclient

Table 5-7 Nrfclient

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
deploymentNrfClientService.envNfNamespace	Namespace in which NSSF is deployed	ocnssf	O		
configmapApplicationConfig.appProfiles	List of NF-Profiles to register to NRF	NA	M	NSSF-Profile is used to register to NRF	List contains only one profile which is of NSSF
configmapApplicationConfig.nrfApiRoot	URL of NRF	NA	M		
nfApiRoot	URL pointing to ingress gateway of NSSF	NA	O		
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

perf-info

Table 5-8 perf-info

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
service_namespace	Namespace in which NSSF is deployed	ocnssf	O		If no value is specified, NSSFs load reported to NRF is always 0.
configmapPerformance.prometheus	Specifies Prometheus server URL	No	http://prometheus-server.prometheus:5802		If no value is specified, NSSFs load reported to NRF is always 0.

Table 5-8 (Cont.) perf-info

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

6
Upgrading NSSF

Upgrading an existing deployment replaces the running containers and pods with new ones. If there is no change in the pod configuration, it will not get replaced. Unless there is a change in the service configuration of a microservice, the service endpoints will remain unchanged (NodePort etc.)

For the parameters that can be configurable, see section Customizing the OCNSSF.

Execute the following command to upgrade an existing NSSF deployment.

```
$ helm upgrade <release> <helm chart> [--version <OCNSSF version>] -
f<ocnssf_customized_values.yaml>
```

- `<release>` could be found in the output of 'helm list' command.
- `<chart>` is the name of the chart in the form of `<repository/ocnssf>`.

Example: `reg-1/ocnssf or cne-repo/ocnssf`

7 Uninstalling NSSF

To delete the NSSF deployment execute the following commands:

To completely delete or remove the NSSF deployment, execute:

```
helm del --purge <helm-release>
```

For example:

```
helm del --purge ocnssf
```

Delete kubernetes namespace

```
kubectl delete namespace <ocnssf kubernetes namespace>
```

For example:

```
kubectl delete namespace ocnssf
```

8 Troubleshooting Information

If you experience issues while using NSSF, refer to the topics below:

Verifying Environment

1. Check if kubectl is installed and working as expected.
2. Check if `kubectl version` command works: This must display the versions of client and server.
3. Check if `$ kubectl create namespace test` command works.
4. Check if `kubectl delete namespace test` command works.
5. Check if Helm is installed and working as expected.
6. Check if `helm version` command works: This must display the versions of client and server.

Debugging of NSSF Installation while Installing using helm

1. If the user is getting the following error:
   failed to parse ocnssf-custom-values-1.3.0.yaml: error converting YAML to JSON: yaml. Then
   a. The `ocnssf-custom-values-1.3.0.yaml` may not be created properly.
   b. The tree structure may not be followed.
   c. There may be tab spaces in the file.

   Verify that the `ocnssf-custom-values-1.3.0.yaml` is proper

   Refer *Network Slice Selection Function (NSSF) Cloud Native Installation Guide .*
2. If there is no error, helm installation will be deployed.
   Helm status can be checked using following command :

   ```
   helm status <helm release name>
   ```

Verifying NSSF Installation

To verify whether the NSSF installation is successful or not, check:

1. Verify NSSF specific pods are working as expected by executing following command
   `kubectl get pods -n <ocnssf_namespace>`

   Check whether all the pods are up and running.

   Sample output:

   ```
   NAME                            READY   STATUS  RESTARTS  AGE
   ocnssf-appinfo-55bcc48477-knc8t             1/1   Running    0     317s
   ocnssf-config-server-5d8c7968fc-d8q9z       1/1   Running    0     317s
   ocnssf-egress-595948cd5-d6hw2               1/1   Running    0     317s
   ```

```
ocnssf-ingress-54c8b668c5-nlcbf            1/1   Running    0      317s
ocnssf-nrf-clientservice-7d9c64f566-z9dnw  1/1   Running    0      317s
ocnssf-nsavailability-8646844976-8pljq     1/1   Running    0      317s
ocnssf-nsconfig-5d755b7865-nh4r9           1/1   Running    0      317s
ocnssf-nsdb-585f7bd7d-bwwjd                1/1   Running    0      317s
ocnssf-nsselection-56674986bf-fmswv        1/1   Running    0      317s
ocnssf-nssubscription-76478c6c84-fdz7f     1/1   Running    0      317s
ocnssf-performance-6d75c7f966-57fgt        1/1   Running    0      317s
```

2. If status of any pod is shown as `ImagePullBackOff` or `ErrImagePull` then it can be due to:
   a. Incorrect ImageName provided in `ocnssf-custom-values-1.3.0.yaml.`
      Then, double check the image name and tags in `ocnssf-custom-values-1.3.0.yaml.`
   b. Docker registry is incorrectly configured.
      Then, check docker registry is properly configured in all master and slave nodes.
3. If RESTARTS count of the pods is continuously increasing, then it can happen due to the following reasons:
   a. MySQL primary and secondary hosts may not be configured properly in `ocnssf-custom-values-1.3.0.yaml`
   b. MySQL servers may not be configured properly according to the pre-installation steps mentioned in *Network Slice Selection Function (NSSF) Cloud Native Installation Guide* .

Debugging General CNE

Execute the command `kubectl get events -n <ocnssf_namespace>`to get all the events related to a particular namespace.

Collecting NSSF Logs

The following commands must be executed to get the logs from nssf specific pods:

1. Fetch the list of all pods by executing `kubectl get pods -n <ocnssf_namespace>`
2. Collect the logs from the pod and redirect to file by executing `kubectl logs <pod_name> -n <ocnssf_namespace> > <Log File>`

Example:

```
kubectl logs ocnssf-nsselection-57cff5665c-skk4l -n ocnssf >
ocnssf_logs1.log
```

A

NSSF Microservices to Port Mapping

Number	NF Service Name	Accesses the DB Tier	Nature of port	Nature of IP	Service Port	Container Port	User	Traffic type	Notes
1	**OCNSSF Ingress G/W**	No	External	Load Balancer	80	8081	5G Peer	Signaling Messages	HTTP2/0 Port (unsecured)
2					443	8443	5G Peer		HTTPS2/0 Port (secured)
3					30080	8081	5G Peer		Static Node Port on demand. Configurable. HTTP2/0 Port (unsecured)
4					30443	8443	5G Peer		Static Node Port on demand, Configurable. HTTPS2/0 Port (secured)
5			Internal	POD IP		9090	Prometheus		
6			Internal	POD IP		9090	Liveliness/ Readiness		
7			Internal		6831				Jaeger Agent port
8	**OCNSSF Egress gateway**	No	Internal	Cluster IP	8080	8080	5G Peer	Signaling Messages	Both for HTTP2/0 Port (unsecured) and HTTPS2/0 Port (secured)
9				POD IP		5701	Intra-Pod Communication		Used internally for hazel-cast
10				POD IP		9090	Prometheus		
11				POD IP		9090	Liveliness/ Readiness		

Number	NF Service Name	Accesses the DB Tier	Nature of port	Nature of IP	Service Port	Container Port	User	Traffic type	Notes
12	**NS Selection**	Yes	Internal	Cluster IP	8080	8080	OCNSSF Ingress GW		
13				POD IP		8080	Prometheus		
14						8080	Liveliness/Readiness		
15				NDB Mysql Service	3306				MYSQL port will be opened by NDB cluster, Microservices are the user of the service. Any change in NDB cluster port needs to be updated here as well.
16	**NS Availability**	Yes	Internal	Cluster IP	8080	8080	OCNSSF Ingress GW		
17				POD IP		8080	Prometheus		
18						8080	Liveliness/Readiness		
19				NDB Mysql Service	3306				MYSQL port will be opened by NDB cluster, Microservices are the user of the service. Any change in NDB cluster port needs to be updated here as well.
20	**NS Subscription**	Yes	Internal	Cluster IP	8080	8080	OCNSSF Ingress GW		

Number	NF Service Name	Accesses the DB Tier	Nature of port	Nature of IP	Service Port	Container Port	User	Traffic type	Notes
21				POD IP		8080	Prometheus		
22						8080	Liveliness/Readiness		
23				NDB Mysql Service	3306				MYSQL port will be opened by NDB cluster, Microservices are the user of the service. Any change in NDB cluster port needs to be updated here as well.
24	**NS Config**	Yes	Internal	Cluster IP	8080	8080	OCNSSF Ingress GW		
25				POD IP		8080	Prometheus		
26						8080	Liveliness/Readiness		
27				NDB Mysql Service	3306				MYSQL port will be opened by NDB cluster, Microservices are the user of the service. Any change in NDB cluster port needs to be updated here as well.
28	**NRF Client**	Yes	Internal	Cluster IP	8080	8080	OCNSSF Ingress GW		
31				POD IP		8080	Liveliness/Readiness		

Number	NF Service Name	Accesses the DB Tier	Nature of port	Nature of IP	Service Port	Container Port	User	Traffic type	Notes
32				NDB Mysql Service	3306				MYSQL port will be opened by NDB cluster, Microservices are the user of the service. Any change in NDB cluster port needs to be updated here as well.

B

Sample values.yaml file

This section provides information about the configurable parameters and values defined in the custom values.yaml template file.

The following sample illustrates the values.yaml file:

```
# Copyright 2019 (C), Oracle and/or its affiliates. All rights reserved.

# This yaml file could be supplied in helm install command when deploying
OCNSSF v1.x.y
#
# e.g. helm install <helm-repo>/ocnssf --name ocnssf --namespace ocnssf -f
<this file>
#
# Compatible with OCNSSF CHART VERSION 1.x.y
# - To turn on logging
#     set the appropriate logging level (one of: OFF, INFO, DEBUG, ERROR,
ALL) in one or more of the following:


global:
  # Docker registry name
  dockerRegistry: ocnrf-registry.us.oracle.com:5000

  # DB credentials
  dbCredSecretName: 'ocnssf-db-creds'

  # Specify type of service - Possible values are :- ClusterIP, NodePort,
LoadBalancer and ExternalName
  type: LoadBalancer

  # Enable or disable IP Address allocation from Metallb Pool
  metalLbIpAllocationEnabled: true

  # Address Pool Annotation for Metallb
  metalLbIpAllocationAnnotation: "metallb.universe.tf/address-pool:
signaling"

  # If Static load balancer IP needs to be set, then set
staticIpAddressEnabled flag to true and provide value for staticIpAddress
  # Else random IP will be assigned by the metalLB from its IP Pool
  staticIpAddressEnabled: false
  staticIpAddress: 10.75.212.60

  # If Static node port needs to be set, then set staticNodePortEnabled
flag to true and provide value for staticNodePort
  # Else random node port will be assigned by K8
  staticNodePortEnabled: true
  staticHttpNodePort: 30075
  staticHttpsNodePort: 30043

  #NRF CLIENT PARAMS
  # Jaeger tracing host
  envJaegerAgentHost: ''
  # Jaeger tracing port
  envJaegerAgentPort: 6831
  # Provide value for NodePort
  nrfClientNodePort: 0
  # Mysql Host
  envMysqlHost: ocnssf-nsdb
  # Mysql Port
  envMysqlPort: '3306'
  # Deployment Specific configuration
  deploymentNrfClientService:
    # Service to be monitored by app-info service
    envNfNamespace: 'ocnssf'
    envNfType: 'nssf'
    # Callback URI to receive Notifications from NRF
    nfApiRoot: http://ocnssf-ingress:80

nsselection:
  replicaCount: 2
  minReplicas: 2
  maxReplicas: 12
  image:
    repository: reg-1:5000
  loglevel: "OFF"
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
  nrf:
    primaryUrl: http://ocnrf.oracle.com:80
    secondaryUrl: http://ocnrf.oracle.com:80
  httpMaxRetries: 0
  homeMcc: "100"
  homeMnc: "101"
  reqnftime: false
  outboundProxy: disabled
  features:
    nrfdiscovery: true
    relevance: true
    candidateResolution: true
  nrfDiscoveryProperties:
    disclimit: 5
  candidateResolutionProperties:
    maxcandidates: 3

nsavailability:
  replicaCount: 2
  minReplicas: 2
  maxReplicas: 12
  image:
    repository: reg-1:5000
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
  loglevel: "OFF"
  maxExpiryDuration: 240
  minExpiryDuration: 0

nsconfig:
  image:
    repository: reg-1:5000
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
  loglevel: "OFF"
  nrf:
    subscription: true
  # URL at which NSSF receives notifications from Nrf. Set when NRF
subscription is turned ON.
  notificationHandlerUrl: http://ocnssf-ingress:80

nrfclient:
  # Microservice level control if specific microservice need to be disabled
  nrf-client:
    # This config map is for providing inputs to NRF-Client
    configmapApplicationConfig:
      # Config-map to provide inputs to Nrf-Client
      # primaryNrfApiRoot - Primary NRF Hostname and Port
      # SecondaryNrfApiRoot - Secondary NRF Hostname and Port
      # retryAfterTime - Default downtime(in Duration) of an NRF detected
to be unavailable.
      # nrfClientType - The NfType of the NF registering
      # nrfClientSubscribeTypes - the NFType for which the NF wants to
subscribe to the NRF.
      # appProfiles - The NfProfile of the NF to be registered with NRF.
      # enableF3 - Support for 29.510 Release 15.3
      # enableF5 - Support for 29.510 Release 15.5
      # renewalTimeBeforeExpiry - Time Period(seconds) before the
Subscription Validity time expires.
      # validityTime - The default validity time(days) for subscriptions.
      # enableSubscriptionAutoRenewal - Enable Renewal of Subscriptions
automatically.
      # acceptAdditionalAttributes - Enable additionalAttributes as part
of 29.510 Release 15.5
      # retryForCongestion - The duration(seconds) after which nrf-client
should retry to a NRF server found to be congested.
      profile: |-
    [appcfg]
    primaryNrfApiRoot=http://ocnrf.oracle.com:80
    secondaryNrfApiRoot=http://ocnrf.oracle.com:80
    retryAfterTime=PT120S
    nrfClientType=NSSF
    nrfClientSubscribeTypes=AMF
    appProfiles=[{"nfInstanceId": "9faf1bbc-6e4a-4454-a507-
aef01a101a06","nfType":"NSSF","nfStatus":"REGISTERED","plmnList":
[{"mcc":"311","mnc":"14"}],"fqdn":"nssf1.lab.oracle.com","interPlmnFqdn":"n
ssf1.lab.oracle.com","ipv4Addresses":
["127.0.0.1","10.0.0.1"],"ipv6Addresses":["::1","::2"],"priority":
5,"load":"20","capacity":"1000","locality":"us-east","amfInfo":
{"amfRegionId":"01","amfSetId":"101","guamiList":[{"plmnId":
{"mcc":"100","mnc":"101"},"amfId":"ABF001"}]},"nfServices":
[{"serviceName":"nssf-
nsselection","nfServiceStatus":"REGISTERED","serviceInstanceId":"123","vers
ions":
[{"apiVersionInUri":"v1","apiFullVersion":"1.15.3.0","expiry":"2019-12-31T2
3:59:59.000+0000"}],"scheme":"http","allowedNfTypes":
["AMF"],"fqdn":"ocnssf-
nsgateway.ocnssf.svc.us.lab.oracle.com","interPlmnFqdn":"ocnssf-
nsgateway.ocnssf.svc.us.lab.oracle.com","ipEndPoints":
[{"ipv4Address":"127.0.0.1","transport":"TCP","port":80}]},
{"serviceName":"nssf-
nsavailability","nfServiceStatus":"REGISTERED","serviceInstanceId":"124","v
ersions":
[{"apiVersionInUri":"v1","apiFullVersion":"1.15.3.0","expiry":"2019-12-31T2
3:59:59.000+0000"}],"scheme":"http","allowedNfTypes":
["AMF"],"fqdn":"ocnssf-
nsgateway.ocnssf.svc.us.lab.oracle.com","interPlmnFqdn":"ocnssf-
nsgateway.ocnssf.svc.us.lab.oracle.com","ipEndPoints":
[{"ipv4Address":"127.0.0.1","transport":"TCP","port":80}]}]}]
    enableF3=true
    enableF5=true
    renewalTimeBeforeExpiry=3600
    validityTime=30
    enableSubscriptionAutoRenewal=true
    acceptAdditionalAttributes=false
    retryForCongestion=5

 # Details of Config-server microservice
 config-server:
  # Mysql Config Server Databse Name
  envMysqlDatabase: ocpm_config_server

 # Details of appinfo microservices
 appinfo:
  debug: true

 # Details of perf-info microservices
 perf-info:
  # Service namespace for perf-info
  service_namespace: ocnssf
  configmapPerformance:
   prometheus: http://prometheus-server.prometheus:5802
```

```
nssubscription:
  replicaCount: 2
  minReplicas: 2
  maxReplicas: 12
  image:
    repository: reg-1:5000
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
  httpMaxRetries: 0
  loglevel: "OFF"
  # oauthTokenRequestEnabled when set true lets Subscription Notifications
to be send with OauthToken
  # As all notifications are send by Egress gateway. oauthClientEnabled in
Egress should also be set true to make this work.
  oauthTokenRequestEnabled: false

ingress-gateway:
  service:
    ssl:
      tlsVersion: TLSv1.2

      privateKey:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        rsa:
          fileName: rsa_private_key_pkcs1.pem
        ecdsa:
          fileName: ec_private_key_pkcs8.pem

      certificate:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        rsa:
          fileName: rsa_apigatewayTestCA.cer
        ecdsa:
          fileName: apigatewayTestCA.cer

      caBundle:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        fileName: caroot.cer

      keyStorePassword:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        fileName: key.txt

      trustStorePassword:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        fileName: trust.txt

      initialAlgorithm: RSA256

  log:
    level:
      root: WARN
      egress: INFO
      oauth: INFO

  # Min replicas to scale to maintain an average CPU utilization
  minReplicas: 1
  # Max replicas to scale to maintain an average CPU utilization
  maxReplicas: 1

  # enable jagger tracing
  jaegerTracingEnabled: false

  openTracing :
    jaeger:
      udpSender:
        # udpsender host
        host: "jaeger-agent.cne-infra"
        # udpsender port
        port: 6831
      probabilisticSampler: 0.5

  #OAUTH CONFIGURATION
  oauthValidatorEnabled: false
  nfType: NSSF
  nfInstanceId: fe7d992b-0541-4c7d-ab84-c6d70b1b01b1
  producerScope: nnssf-nsselection,nnssf-nsavailability
  allowedClockSkewSeconds: 0
  nrfPublicKeyKubeSecret: nrfpublickeysecret
  nrfPublicKeyKubeNamespace: ocnssf
  validationType: strict
  producerPlmnMNC: 123
  producerPlmnMCC: 346

  #######################################################################
  # To Initialize SSL related infrastructure in init/update container
  # initssl: true
  #Server Configuration for http and https support

  enableIncomingHttp: true
  enableIncomingHttps: false
  enableOutgoingHttps: false

  #TLS certificate reload for https
  ingressGwCertReloadEnabled: true
  #######################################################################
  serviceMeshCheck: false

  #CnCoam bug fix
  cnccIamEnabled: false
```

```
  #IAM configuration
  identityAccessMgt:
    uri: http://demo.iam:30024
    path: /cncc/auth
    realm: cncc
    clientId: api-gateway

  #Jetty Client settings
  maxConnectionsQueuedPerDestination: 1024
  maxConnectionsPerDestination: 4
  maxConnectionsPerIp: 4
  connectionTimeout: 10000 #(ms)

  #Rate limiting configuration
  rateLimiting:
    enabled: false
  routeRateLimiting:
    enabled: true
  globalIngressRateLimiting:
    enabled: true
    duration: 60 # in seconds
    burstCapacity: 4
    refillRate: 2

egress-gateway:
  #Enabling it for ougoing https request
  enableOutgoingHttps: false

  #Enable this if loadbalancing is to be done by egress instead of K8s
  K8ServiceCheck: false

  #SCP Configuration for egress gateway
  scp:
    # Default scheme applicable when 3gpp-sbi-target-apiroot header is
missing
    scpDefaultScheme: https
    # Change this to false when scp integration is not required
    scpIntegrationEnabled: false
    # Set this flag to true if re-routing to multiple SCP instances is to
be enabled.
    scpRerouteEnabled: false
    instances:
      http:
      - host: localhost
        port: 101
        apiPrefix: "/"
      - host: localhost
        port: 102
        apiPrefix: "/"
      - host: 10.75.224.7
        port: 32070
        apiPrefix: "/"
      https:
      - host: localhost
        port: 4431
        apiPrefix: "/" # Change this value to corresponding prefix "/" is
not expected to be provided along.
      - host: localhost
        port: 4432
        apiPrefix: "/"
      - host: 10.75.224.109
        port: 30570
        apiPrefix: "/"

  #Enabling this will make the service type default to ClusterIP
  headlessServiceEnabled: false

  log:
    level:
      root: WARN
      egress: INFO
      oauth: INFO

  service:
    # Specify type of service - Possible values are :- ClusterIP,
NodePort, LoadBalancer and ExternalName
    type: ClusterIP
    ssl:
      tlsVersion: TLSv1.2

      privateKey:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        rsa:
          fileName: rsa_private_key_pkcs1.pem
        ecdsa:
          fileName: ec_private_key_pkcs8.pem

      certificate:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        rsa:
          fileName: rsa_apigatewayTestCA.cer
        ecdsa:
          fileName: apigatewayTestCA.cer

      caBundle:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        fileName: caroot.cer

      keyStorePassword:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        fileName: key.txt

      trustStorePassword:
        k8SecretName: accesstoken-secret
        k8NameSpace: ocnssf
        fileName: trust.txt
      initialAlgorithm: RSA256

  #globalretry can be enabled only when scpRerouteEnabled flag is set to 
true.
  globalretry:
    enabled: false
    retries: 2

  routesConfig:
  - id: scp_path1
    uri: https://request.uri
    path: /nef/**
    order: 1
    filterName1: ScpFilter
    filterName2:
      name: ScpRetry
      retries: 1
      methods: GET, POST, PUT, DELETE, PATCH
      statuses: BAD_REQUEST, INTERNAL_SERVER_ERROR, BAD_GATEWAY, NOT_FOUND
  - id: scp_path2
    uri: https://dummy.dontchange1
    path: /npcf/**
    order: 2
    filterName1: ScpFilter
    filterName2:
      name: ScpRetry
      retries: 1
      methods: GET, POST, PUT, DELETE, PATCH
      statuses: BAD_REQUEST, INTERNAL_SERVER_ERROR, BAD_GATEWAY, NOT_FOUND
  - id: scp_path3
    uri: https://dummy.dontchange2
    path: /nxyz/**
    order: 3
    filterName1: ScpFilter
    filterName2:
      name: ScpRetry
      retries: 1
      methods: GET, POST, PUT, DELETE, PATCH
      statuses: BAD_REQUEST, INTERNAL_SERVER_ERROR, BAD_GATEWAY, NOT_FOUND
  - id: egress_iwf
    uri: egress://test.com
    path: /niwf/**
    order: 4

  # Min replicas to scale to maintain an average CPU utilization
  minReplicas: 1
  # Max replicas to scale to maintain an average CPU utilization
  maxReplicas: 1

  nrfAuthority: ocnrf.oracle.com:80
  nfType: NSSF
  nfInstanceId: fe7d992b-0541-4c7d-ab84-c6d70b1b01b1
  oauthClientEnabled: false
  consumerPlmnMNC: 101
```

```
consumerPlmnMCC: 100
#Jetty bean name
#when http enabled -> ''
#when https enabled -> jettysClient
httpClientBean: ''

#jetty client configuration
maxConnectionsQueuedPerDestination: 1024
maxConnectionsPerIp: 4
connectionTimeout: 10000 #(ms)

egressGwCertReloadEnabled: true

#enable jagger tracing
jaegerTracingEnabled: false

openTracing:
  jaeger:
    udpSender:
      # udpsender host
      host: "occne-tracer-jaeger-agent.occne-infra"
      # udpsender port
      port: 6831
    probabilisticSampler: 0.5

# Flag to enable rate limiting for "notification" type of messages.
notificationRateLimit:
  enabled: false
  duration: 60
  bucketCapacity: 4
  refillRate: 2
```