Oracle® Communications

Network Slice Selection Function (NSSF) Cloud Native Installation Guide

Release 1.4
F33536-01
July 2020

Oracle Communications Network Slice Selection Function (NSSF) Cloud Native Installation Guide, Release 1.4

F33536-01

Contents

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.

What's New in This Guide

New and Updated Features and their Configurations in Release 1.4:

- Support for HTTP Option method.
- Helm parameters are updated to section NSSF Configurable Parameters.
- Configurable parameters are added at NS-Availability Microservice at section NSSF Configurable Parameters.
- Helm 2/ helm 3 commands are added for installation to the section NSSF Installation and for uninstallation to the section Uninstalling NSSF.

List of Tables

1 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). Access and Mobility Management Function (AMF) can retrieve NRF, NSI ID, and target AMFs as part of UE initial registration and Packet Data Unit (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 1-1 Acronyms and Terminologies

Field	Description
AMF	Access and Mobility Management Function
Allowed NSSAI	NSSAI provided by the Serving PLMN during 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

Table 1-1 (Cont.) Acronyms and Terminologies

Field	Description
NF service	A functionality exposed by a NF through a service based interface and consumed by other authorized NFs.
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

2
Installing NSSF

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

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 table above is not installed in CNE, install the software.
> If OCCNE is the platform, 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.

Client Machine 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 Installation Sequence

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

1. **Installation Preparation:** This includes downloading the required files and loading the files to the system.
2. **Configure the `custom_values.yaml` file:** This step includes configuring the following based on the deployment:
   - Repository path
   - Primary and Secondary node

- 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:** In this step if the services and pods are up and running is verified.

Docker Images for NSSF

Following are the NSSF images:

Table 2-1 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 to chapter *OCCNE Docker Image Registry Configuration* in *OCCNE Installation Guide*.

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-1.4.0.0.0.tgz
   ```
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:

- NSSF Docker Images File
  `ocnssf-images-1.4.0.tar`
- Helm File
  `ocnssf-1.4.0.tgz`
- 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 load the tarball to system:

   ```
   docker load --input ocnssf-images-1.4.0.tar
   ```

5. **Check if all the images are loaded:**

   Execute the following command to check whether alll the images are loaded:

   ```
   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:**

   Execute the following command to untar the helm files:

   ```
   tar -xvzf ocnssf-1.4.0.tgz
   ```

8. **Download the Network Slice Selection Function (NSSF) Custom Template ZIP file:**

   Download the **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 **Network Slice Selection Function (NSSF) Custom Template** link to download the zip file.
- Unzip the template to get the following files:
  - `ocnssf-custom-values-1.4.0.yaml`
  - `onssfDashboard.json`

NSSF Installation

This section includes information about NSSF deployment.

Following are the parameters and definitions used during NSSF deployment:

Table 2-2 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 microservices 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 and 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.

NSSF Deployment

1. **Create customized `ocnssf-custom-values-1.4.0.yaml` file** :

   Create the customized `ocnssf-custom-values-1.4.0.yaml` with the required input parameters.
   To configure the `ocnssf-custom-values-1.4.0.yaml`, refer to Customizing NSSF

   or,
   The `ocnssf-custom-values-1.4.0.yaml` template can be downloaded from OHC.

   Download the package `ocnssf-custom-configTemplates-1.4.0.0.0.zip` and Unzip to get `ocnssf-custom-values-1.4.0.yaml` file.

2. **Go to the unzipped OCNSSF package:**
   Go to the following directory:

   ```
   cd OCNSSF-pkg-1.4.0.0.0
   ```

3. **Deploy OCNSSF:**

   Execute the following command:
   For helm 2 based:

   ```
   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-1.4.0.yaml
```

Example:

```
helm install ocnssf/ --name ocnssf --namespace ocnssf -f ocnssf-
custom-values-1.4.0.yaml
```

For helm 3 based:

```
helm install <helm-release> ocnssf/ --namespace <k8s namespace> f
<ocnssf_customized_values.yaml>

Example:
helm install ocnssf ocnssf/ --namespace ocnssf -f ocnssf-custom-
values-1.4.0.yaml
```

4. **Check status of the deployment:**
   Execute the following command to check the status of the deployment:

   `helm status --name <helm-release>`
   Example: `helm status --name ocnssf`

5. **Check status of the services:**

   Execute the following command to check the status of services:
   `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.

Sample output:

```
NAME                 TYPE         CLUSTER-IP  EXTERNAL-IP
PORT(S)                AGE
ocnssf-appinfo               ClusterIP    10.98.169.9    <none>
5906/TCP                     44h
ocnssf-egress                ClusterIP    10.107.165.12  <none>
8080/TCP                     44h
ocnssf-ingress               LoadBalancer 10.107.131.252 <pending>
80:30075/TCP                 44h
ocnssf-nrf-client-nfdiscovery NodePort     10.104.86.218  <none>
5910:30598/TCP,5805:31734/TCP 44h
ocnssf-nrf-client-nfmanagementNodePort     10.104.31.40   <none>
5910:30847/TCP,5805:31488/TCP 44h
ocnssf-nsavailability        NodePort     10.97.145.148  <none>
5745:32218/TCP               44h
ocnssf-nsconfig              NodePort     10.104.57.232  <none>
5755:30225/TCP               44h
```

```
ocnssf-nsdb                    ClusterIP    10.109.80.141  <none>
3306/TCP                       44h
ocnssf-nsselection             NodePort     10.102.208.48  <none>
5745:31302/TCP,4546:32109/TCP  44h
ocnssf-nssubscription          NodePort     10.107.153.91  <none>
5745:30302/TCP,4546:31375/TCP  44h
ocnssf-ocpm-config             ClusterIP    10.103.10.228  <none>
5807/TCP,9000/TCP              44h
ocnssf-performance             NodePort     10.107.40.119  <none>
5905:31553/TCP                 44h
```

6. **Check status of the pods:**

   Execute the following command to status of the pods:
   `kubectl get pods -n <k8s namespace>`

   Status column of all the pods must indicate 'Running'.

   Ready column of all the pods must be n/n, where n is number of containers in the pod.

   Example:

   ```
   kubectl get pods -n ocnssf
   ```

   Sample output :

   ```
   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-client-nfdiscovery-cf48cd8d8-l4q2q    1/1  Running   0
   18m
   ocnssf-nrf-client-nfdiscovery-cf48cd8d8-vmt5v    1/1  Running   0
   18m
   ocnssf-nrf-client-nfmanagement-7db4598fbb-672hc  1/1  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
   ```

NSSF Pre-deployment Configuration

This section describes about the various NSSF pre-deployment configuration steps. 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
6. Create Role and Role Binding

Verify and Create Kubernetes Namespace

his section explains how a user can verify the existence of a required namespace in the system. If the namespace does not exist, the 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, create the namespace by executing the 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 at the time.
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 the user does not exist, create an OCNSSF network function user by executing the following queries:
   a. Execute`$ SELECT User FROM mysql.user;` to list the users.
   b. If the 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 if OCNSSF network function database already exists. If it does not exist, create an OCNSSF network function database and provide permissions to OCNSSF username created in the previous step by executing the following commands:
   a. Execute `$ show databases;` to check if database exists.
   b. If database does not exist, 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 namespace 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
   ```

Create Role and Role Binding

This section describes about how to create role and do role binding with service account to access secret in namespace.

1. **Create role**

   Create a file "role.yaml" with following content.

   role.yaml

   ```
   kind: Role
   apiVersion: rbac.authorization.k8s.io/v1
   metadata:
     name: ocnssf-role
   rules:
   ```

```
  - apiGroups: [""] # "" indicates the core API group
    resources: ["secrets"]
    verbs: ["get", "list"]
```

Run the following command to create role:

kubectl apply -f role.yaml -n ocnssf

2. **Role Binding**
   Bind the role with the default service account of the namespace.

   rolebinding.yaml

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: ocnssf-rolebinding
subjects:
  - kind: ServiceAccount
    name: default
    namespace: ocnssf # use release reference here
roleRef:
  kind: Role
  name: ocnssf-role
  apiGroup: rbac.authorization.k8s.io
```

Run the following command to create rolebinding with default service:

kubectl apply -f rolebinding.yaml -n ocnssf

3
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.

Customizing NSSF

The NSSF deployment is customized by overriding the default values of various configurable parameters in the ocnssf-custom-values-1.4.0.yaml file.

To customize the ocnssf-custom-values-1.4.0.yaml file as per the required parameters:

1. Go to the Oracle Help Center (OHC) Web site: https://docs.oracle.com
2. Navigate to Industries >Communications >Cloud Native Core >Release 2.2.1
3. Click the Network Slice Selection Function (NSSF) Custom Template link to download the zip file.
4. Unzip the file to get ocnssf-custom-configTemplates-1.4.0.0.0 file that contains the ocnssf-custom-values-1.4.0.yaml. This file is used during installation.
5. Customize the ocnssf-custom-values-1.4.0.yaml file.
6. Save the updated ocnssf-custom-values-1.4.0.yaml file in the helm chart directory.

The sample ocnssf-custom-values-1.4.0.yaml file created based on all the parameters described in the Configurable Parameters section .

NSSF Configurable Parameters

NS-Selection

Table 3-1 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.
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.

Table 3-1 (Cont.) NS-Selection

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
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.
candidateResolutionProperties.maxcandidates:	Maximum number of candidate AMFs	3	Mandatory when features.candidateResolutioni s 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

Table 3-1 (Cont.) NS-Selection

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
mysql.secondary.host	Secondary MYSQL Host IP or Hostname	*o*cnssf-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 3-2 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.

Table 3-2 (Cont.) NS-Availability

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
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
contentEncodingEnabled	To enable or disable response gzip compression	True	O	True or False	If value is True content-encoding (json to gzip) is enabled at server side (ocnssf). If value is false content-encoding is not enabled.
compressionMinimumResponseSize	Minimum response size required for compression to happen (size is in bytes).	1024	O	Any value	Signifies the minimum size the response has to be in order for it to be compressed (and sent as gzip)
maxRequestSize	Maximum limit for request size	1MB	O	Any Value	If request is larger than "maxRequestSize", then HTTP 413 (Request Entity Too Large error) response is sent back.

NS-Config

Table 3-3 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	

Table 3-3 (Cont.) NS-Config

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Notes
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

NS-Subscription

Table 3-4 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.

Table 3-4 (Cont.) NS-Subscription

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Note
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
mysql.port	Port of MYSQL Database	3306	M	Port of MySQL Database	
image.repository	Full Image Path		M	Full image path of image	

Table 3-4 (Cont.) NS-Subscription

Helm Parameter	Description	Default Value	Mandatory (M)/ Optional (O)	Accepted Values	Note
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

Common Micro Services

Ingress Gateway

Table 3-5 Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
global.dockerRegistry	Name of the Docker registry which hosts Ingress docker images.	ocnrf-registry.us.oracle.com:5000	Yes	This is the registry which has docker images. Change this value if there is a need.
global.type	type of service	LoadBalancer	Yes	Possible values are :- ClusterIP, NodePort, LoadBalancer and ExternalName
global.serviceAccountName	Service Account name	''	No	
global.metalLbIpAllocationEnabled	Enable or disable IP Address allocation from Metallb Pool	true	No	
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		

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
global.publicHttpSignalingPort	Http Signaling port	80	Yes	
global.publicHttpsSignallingPort	Https Signaling port	443	Yes	
global.staticNodePortEnabled	Node Port Enabled	true	No	
global.staticHttpNodePort	Http Node Port	30075	Yes	
global.staticHttpsNodePort	Https Node Port	30043	Yes	
global.configServerFullNameOverride	This parameter is for the usage of policy teams. Other teams can ignore this parameter.		No	
enableOutgoingHttps	Enabling it for outgoing https request	false	Yes	Change it to true for enabling https for outgoing requests.
enableIncomingHttp	Enabling it for incoming http request	false	Yes	
enableIncomingHttps	Enabling it for incoming https request	true	Yes	
enablehttp1	Enable it for http1.1	false	No	Change it to true to enable
dnsRefreshDelay	Dns Refresh Delay in milli-seconds	120000	No	
oauthValidatorEnabled	Oauth Validator Enabled	false	Yes	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	Yes (If jaegerTracingEnabled is true)	
openTracing.jaeger.udpSender.port	Jaeger Port	6831	Yes (If jaegerTracingEnabled is true)	
openTracing.jaeger.probabilisticSampler		0.5	Yes (If jaegerTracingEnabled is true)	
nfType	NFType of service producer.	Value to be updated accordingly	Yes (When oauthValidatorEnabled)	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
nfInstanceId:	NF InstanceId of service producer.	Value to be updated accordingly	Yes (When oauthValidatorEnabled)	
producerScope:	Comma-separate list of services hosted by service producer.	Value to be updated accordingly	Yes (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	Yes (When oauthValidatorEnabled)	
nrfPublicKeyKubeSecret	Name of the secret which stores the public key(s) of NRF.	Value to be updated accordingly	Yes (When oauthValidatorEnabled)	
nrfPublicKeyKubeNamespace	Namespace of the NRF publicKey Secret	Value to be updated accordingly	Yes (When oauthValidatorEnabled)	
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 does not contain "Authorization" header, validation will be ignored.	Value to be updated accordingly	Yes (When oauthValidatorEnabled)	
producerPlmnMNC	MNC of service producer.	Value to be updated accordingly	No	
producerPlmnMCC	MCC of service producer.	Value to be updated accordingly	No	
cncc.enabled	CNCC Identity-Access-Management(IAM).	False	No	Change it to true if required.
cncc.core.sessionTimeoutSeconds	Session Timeout Value in Seconds. Default: 1800, Minimum: 300, Maximum: 7200	1800	No	
ccclamEnabled	CNCC Identity-Access-Management (IAM)	false	No	Change it to true if required
ingressGwCertReloadEnabled		true	No	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
rateLimiting.enabled	Ratelimiting feature enabled	false	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)	yes (if globalIngressRateLimiting.enabled)	
globalIngressRateLimiting.burstCapacity	Holds maximum number of tokens in the bucket for the given duration.	1	yes (if globalIngressRateLimiting.enabled)	
globalIngressRateLimiting.refillRate	Number of tokens to be added to the bucket for the given duration	1	yes (if globalIngressRateLimiting.enabled)	
identityAccessMgt.uri	Identity access management uri		yes (if cnccIamEnabled)	
identityAccessMgt.path	Identity access management path		yes (if cnccIamEnabled)	
identityAccessMgt.realm	Identity access management realm		yes (if cnccIamEnabled)	
identityAccessMgt.clientId	Identity access management client id		yes (if cnccIamEnabled)	
iam.uri The section name is changed to iam	Identity access management uri		yes (if cnccIamEnabled)	
iam.path	Identity access management path		yes (if cnccIamEnabled)	
iam.realm	Identity access management realm		yes (if cnccIamEnabled)	
iam.clientId	Identity access management client id		yes (if cnccIamEnabled)	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
pingDelay	Delay between pings in seconds. When set to <=0,ping is disabled	60	Yes	PING frame can be scheduled at Ingress-gateway to maintain connection between Ingress-gateway and backend micro-services even if the connection is idle.
cfgServer.enabled	Config server switch. For the usage of Policy teams. For other NF's this has to be left false	false	No	
publicHttpSignalingPort	Http Signalling port	80	Yes	
publicHttpsSignallingPort	Https Signalling port	443	Yes	
ssl.privateKey.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.privateKey.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.privateKey.rsa.fileName	rsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.privateKey.ecdsa.fileName	ecdsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.rsa.fileName	rsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.ecdsa.fileName	ecdsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
ssl.caBundle.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.caBundle.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.caBundle.rsa.fileName	rsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.keyStorePassword.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.keyStorePassword.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.keyStorePassword.fileName	File name that has password for keyStore	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.trustStorePassword.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.trustStorePassword.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.trustStorePassword.fileName	File name that has password for trustStore	n/a	Yes (If enableIncomingHttp is true otherwise No)	
publicHttpSignalingPort	Http Signalling port	80	Yes	
publicHttpsSignallingPort	Https Signalling port	443	Yes	
ssl.privateKey.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.privateKey.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
ssl.privateKey.rsa.fileName	rsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.privateKey.ecdsa.fileName	ecdsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.rsa.fileName	rsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.certificate.ecdsa.fileName	ecdsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.caBundle.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.caBundle.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.caBundle.rsa.fileName	rsa private key file name	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.keyStorePassword.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.keyStorePassword.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.keyStorePassword.fileName	File name that has password for keyStore	n/a	Yes (If enableIncomingHttp is true otherwise No)	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
ssl.trustStorePassword.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.trustStorePassword.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableIncomingHttp is true otherwise No)	
ssl.trustStorePassword.fil eName	File name that has password for trustStore	n/a	Yes (If enableIncomingHttp is true otherwise No)	
uri	Service name of the internal microservice of this NF		Yes	
id	id of the route		Yes	
path	Provide the path to be matched.		Yes	
order	Provide the order of the execution of this route.		Yes	
methodRateLimiting.burstCapacity[0]	burstCapacity		Yes (if routeRateLimiting.enabled)	
methodRateLimiting.refillRate[0]	Refill rate		Yes (if routeRateLimiting.enabled)	
methodRateLimiting.duration[0]	Duration		Yes (if routeRateLimiting.enabled)	
methodRateLimiting.method[0]	Method on which ratelimiting is applicable		Yes (if routeRateLimiting.enabled)	
image.name	Image name of ingress gateway	ocingress_gateway	No	
image.tag	Image Tag name of ingress gateway	1.6.2	No	
image.pullPolicy	Image Pull Policy	Always	No	
initContainersImage.name	Image name of initContainer	configurationinit	No	
initContainersImage.tag	Image tag name of initContainer	1.1.1	No	
initContainersImage.pullPolicy	Image Pull Policy	Always	No	
updateContainersImage.name	Image name of updateContainer	configurationupdate	No	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
updateContainersImage.tag	Image tag name of updateContainer	1.1.1	No	
updateContainersImage.pullPolicy	Image Pull Policy	Always	No	
fullnameOverride	Label to override name of api-gateway micro-service name	ingress	Yes	
serviceMeshCheck	Load balancing will be handled by Ingress gateway, if true it would be handled by serviceMesh	false	Yes	
cipherSuites	Supported Cipher Suites in Ingress	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	

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
maxRequestsQueuedPerDestination	Jetty Client Settings	1024	No	
maxConnectionsPerDestination	Jetty Client Settings	4 (This will be used when `serviceMeshCheck` is enabled)	No	
maxConnectionsPerIp	Jetty Client Settings	4	No	
connectionTimeout	Jetty Client Settings	10000	No	
ingressGwCertReloadPath		`/ingress-gw/certificate/reload`	No	
ssl.tlsVersion	TLS Version	TLSv1.2	Yes	
ssl.initialAlgorithm		RSA256	Yes	ES256 can also be used, but corresponding certificates need to be used.
resources.limits.cpu	CPU Limit	2		
resources.limits.memory	Memory Limit	4Gi		
resources.limits.initServiceCpu	Init Container CPU Limit	1		
resources.limits.updateServiceCpu	Update Container CPU Limit	1		
resources.limits.initServiceMemory	Init Container Memory Limit	1Gi		
resources.limits.updateServiceMemory	Update Container Memory Limit	1Gi		
resources.requests.cpu	CPU for requests	1		
resources.requests.memory	Memory for requests	2Gi		

Table 3-5 (Cont.) Ingress Gateway

Name	Description	Default Value	Mandatory	Notes
resources.requests.initServiceCpu	Init Container CPU for requests	1		
resources.requests.updateServiceCpu	Update Container CPU for requests	1		
resources.requests.initServiceMemory	Init Container Memory for requests	1Gi		
resources.requests.updateServiceMemory	Update Container Memory for requests	1Gi		
resources.target.averageCpuUtil		80		
minReplicas	Min replicas to scale to maintain an average CPU utilization	2	Yes	
maxReplicas	Max replicas to scale to maintain an average CPU utilization	5	Yes	
log.level.root	Log level for root logs	WARN	No	
log.level.ingress	Log level for ingress logs	INFO	No	
log.level.oauth	Log level for oauth logs	INFO	No	
ports.containerPort	ContainerPort represents a network port in a single container	8081	No	
ports.containersslPort		8443	No	
actuatorPort	ActuatorPort	9090	No	

Egress Gateway

Table 3-6 Egress Gateway

Name	Description	Default Value	Mandatory	Notes
global.appinfoServiceEnable	Enabled to get RBAC permission for k8s apiserver communication	true	Yes	

Table 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
global.dockerRegistry	Name of the Docker registry which hosts Egress docker images.	ocnrf-registry.us.oracle.com:5000	Yes	Ideally this is the registry which has docker images. Change this value if there is a need.
global.serviceAccountName	Service Account Name	''	No	
serviceEgressGateway.port		8080	No	
serviceEgressGateway.sslPort	SSL Port	8442	No	
serviceEgressGateway.actuatorPort	Actuator Port	9090	No	
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	
scp.scpDefaultScheme	Default scheme applicable when 3gpp-sbi-target-apiroot header is missing	https	No	
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	
scp.instances.http[0].host	First Scp instance HTTP IP/FQDN	NA	Yes(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	Yes(If "scp.scpIntegrationEnabled" is set to true.)	

Table 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	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].host	First Scp instance HTTPS IP/FQDN	NA	Yes(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	Yes(if "scp.scpIntegrationEnabled" is set to true.)	
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.
headlessServiceEnabled	Enabling this will make the service type default to ClusterIP	false	No	

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 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
log.level	Log level	DEBUG	No	
jaegerTracingEnabled	Enable jaeger tracing	false	No	Change it to true if needed.
openTracing.jaeger.udpSender.host	Jaeger Host	jaeger-agent.cne-infra	Yes (If jaegerTracingEnabled is true)	
openTracing.jaeger.udpSender.port	Jaeger Port	6831	Yes (If jaegerTracingEnabled is true)	
openTracing.jaeger.probabilisticSampler		0.5	Yes (If jaegerTracingEnabled is true)	
nrfAuthority	NRF's ${HOSTNAME}:{PORT}	Modify the field with actual value, required if oAuth is enabled.	Yes	
nfType	NFType of service consumer.	Modify the field with actual value , required if oAuth is enabled.	Yes	
nfInstanceId:	NF InstanceId of Service Consumer.	Modify the field with actual value, required if oAuth is enabled.	Yes	
oauthClientEnabled:	Flag to enable or disable oauth client. If not modified, Default value 'false' will be defaulted.	false	No	Change it to true to enable oAuth
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 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
maxRequestsQueuedPerDestination	jetty client configuration	1024	No	
maxConnectionsPerIp	Max Connections allowed per Ip	4	No	
connectionTimeout	Connection timeout in milliseconds	1000	No	
egressGwCertReloadEnabled		true	No	
notificationRateLimit.enabled	Flag to enable rate limiting for "notification" type of messages.	false	No	
notificationRateLimit.duration	Iterations of time duration(In seconds) for which bucketCapacity and refillRate are reset.		Yes(If notificationRateLimit.enabled is set to true)	
notificationRateLimit.bucketCapacity	Holds maximum number of tokens in the bucket for the given duration.		Yes(If notificationRateLimit.enabled is set to true)	
notificationRateLimit.refillRate	Number of tokens to be added to the bucket for the given duration		Yes(If notificationRateLimit.enabled is set to true)	
type	type of service	ClusterIP Possible values are ClusterIP, NodePort, LoadBalancer and ExternalName	Yes	
ssl.privateKey.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.privateKey.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.privateKey.rsa.fileName	rsa private key file name	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.privateKey.ecdsa.fileName	ecdsa private key file name	n/a	Yes (If enableOutgoingHttps is true otherwise No)	

Table 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
ssl.certificate.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.certificate.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.certificate.rsa.fileName	rsa private key file name	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.certificate.ecdsa.fileName	ecdsa private key file name	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.caBundle.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.caBundle.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.caBundle.rsa.fileName	rsa private key file name	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.keyStorePassword.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.keyStorePassword.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.keyStorePassword.fileName	File name that has password for keyStore	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.trustStorePassword.k8SecretName	Name of the privatekey secret	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
ssl.trustStorePassword.k8NameSpace	Namespace of privatekey	n/a	Yes (If enableOutgoingHttps is true otherwise No)	

Table 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
ssl.trustStorePassword.fileName	File name that has password for trustStore	n/a	Yes (If enableOutgoingHttps is true otherwise No)	
resources.limits.cpu	CPU Limit	2		
resources.limits.memory	Memory Limit	4Gi		
resources.limits.initServiceCpu	Init Container CPU Limit	1		
resources.limits.updateServiceCpu	Update Container CPU Limit	1		
resources.limits.initServiceMemory	Init Container Memory Limit	1Gi		
resources.limits.updateServiceMemory	Update Container Memory Limit	1Gi		
resources.requests.cpu	CPU for requests	1		
resources.requests.memory	Memory for requests	2Gi		
resources.requests.initServiceCpu	Init Container CPU for requests	1		
resources.requests.updateServiceCpu	Update Container CPU for requests	1		
resources.requests.initServiceMemory	Init Container Memory for requests	1Gi		
resources.requests.updateServiceMemory	Update Container Memory for requests	1Gi		
resources.target.averageCpuUtil		80		
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	2		

Table 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	5		
globalretry.enabled	Can be set to true if Scp re-route feature (`scpRerouteEnabled`) is enabled.	false	No	
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.		Yes (If `"routesConfig[0].filterName2.retries"` is not defined)	
routesConfig[0].id	id of the route		Yes	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		Yes	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.		Yes	
routesConfig[0].order	Provide the order of the execution of this route.		Yes	
routesConfig[0].filterName1	Provide filtername as "ScpFilter"		Yes (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"		Yes (If scpRerouteEnabled is true)	With out FilterName1 , it is not possible to configure FilterName2.name

Table 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
routesConfig[0].filterName2.retries	Number of re-routes to be attempted to alternate SCP instances if request matches this route's path.		Yes (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.		Yes (If scpRerouteEnabled is true)	
routesConfig[0].filterName2.statuses	The type response error codes on which the re-route need to be attempted.		Yes (If scpRerouteEnabled is true)	
serviceEgressGateway.port	Internal port on which egress gateway is running for HTTP2	No	8080	Change this value if there is any specific need.
serviceEgressGateway.sslPort	Internal port on which egress gateway is running for HTTPS	No	8442	Change this value if there is any specific need.
deploymentEgressGateway.image	Image name of egress gateway	No	ocegress_gateway	N/A
deploymentEgressGateway.imageTag	Image Tag name of egress gateway	No	1.6.1	N/A
deploymentEgressGateway.pullPolicy	Pull Policy of Image	No	Always	N/A
initContainersImage.name	Image name of initContainer	No	configurationinit	N/A
initContainersImage.tag	Image tag name of initContainer	No	1.1.1	N/A
initContainersImage.pullPolicy	Pull Policy of Image	No	Always	N/A
updateContainersImage.name	Image name of updateContainer	No	configurationupdate	N/A
updateContainersImage.tag	Image tag name of updateContainer	No	1.1.1	N/A

Table 3-6 (Cont.) Egress Gateway

Name	Description	Default Value	Mandatory	Notes
updateContainersImage.pullPolicy	Pull Policy of Image	No	Always	N/A
httpClientBean	To be used when oAuth is enabled. when https is enabled then it should be jettysClient , when https is disabled then it can left as ''	Yes	jettysClient	#Jetty bean name #when http enabled -> '' #when https enabled -> jettysClient
egressGwCertReloadEnabled	Egress GW Certificates Reload Enabled	No	true	N/A
jaegerTracingEnabled	JaegerTracing Enabled	No	false	N/A
ssl.tlsVersion	TLS Version	TLSv1.2	Yes	
initialAlgorithm		RSA256	Yes	ES256 can also be used, but corresponding certificates need to be used.

Nrfclient

Table 3-7 Nrfclient

Parameter	Description	Default value	Mandatory (M)/ Optional (O)	Range or Possible Values (If applicable)	Notes
deployment NrfClientService.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		

Table 3-7 (Cont.) Nrfclient

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

perf-info

Table 3-8 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.
image.repository	Full Image Path		M	Full image path of image	
log.level	Logging level	INFO	O	INFO, DEBUG, FATAL, ERROR, WARN	Logging level

4
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 are 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`

5
Uninstalling NSSF

To delete the NSSF deployment, execute the following commands:

To completely delete or remove the NSSF deployment:

For **helm 2** parameters:

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

Example: `helm del --purge ocnssf`

For **helm 3** parameters:

```
helm uninstall <helm-release> --namespace <ocnssf kubernetes namespace>
```

Example: `helm uninstall ocnssf --namespace ocnssf`

To delete kubernetes namespace:

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

Example: `kubectl delete namespace ocnssf`

6
Troubleshooting Information

This section provides information to troubleshoot the common error which can be encountered during the installation and upgrade of NSSF:

Generic Checklist

The following sections provide generic checklist for troubleshooting tips.

Deployment related tips

- Are OCnssf deployment, pods and services created, running and available?

  Execute following the command:

  ```
  # kubectl -n <namespace> get deployments,pods,svc
  ```

  Inspect the output, check the following columns:

  – AVAILABLE of deployment
  – READY, STATUS and RESTARTS of po
  – PORT(S) of service

- Is the correct image used and the correct environment variables set in the deployment? Execute following the command:

  ```
  # kubectl -n <namespace> get deployment <deployment-name> -o yaml
  ```

- Inspect the output, check the environment and image.

  ```
  # kubectl -n nssf-svc get deployment ocnssf-nfregistration -o yaml
  apiVersion: extensions/v1beta1
  kind: Deployment metadata:
                annotations:
  ```

```
              deployment.kubernetes.io/revision: "1"
            kubectl.kubernetes.io/last-applied-configuration:
|
            {"apiVersion":"apps/v1","kind":"Deployment","metadata":
            {"annotations":{},"name":"ocnssf-
nfregistration","namespace":"nssfsvc"},
            "spec":{"replicas":1,"selector":{"matchLabels":
{"app":"ocnssf-nfregistration"}},
            "template":{"metadata": {"labels":{"app":"ocnssf-
nfregistration"}},
            "spec": {"containers":[{"env":
[{"name":"MYSQL_HOST","value":"mysql"},
            {"name":"MYSQL_PORT","value":"3306"},
            {"name":"MYSQL_DATABASE","value":"nssfdb"},

{"name":"nssf_REGISTRATION_ENDPOINT","value":"ocnssfnfregistration"}
,

{"name":"nssf_SUBSCRIPTION_ENDPOINT","value":"ocnssfnfsubscription"}
,
            {"name":"NF_HEARTBEAT","value":"120"},
            {"name":"DISC_VALIDITY_PERIOD","value":"3600"}],
            "image":"dsrmaster0:5000/ocnssfnfregistration:latest",

"imagePullPolicy":"Always","name":"ocnssfnfregistration","ports":
            [{"containerPort":8080,"name":"server"}]}]}}}}
            creationTimestamp: 2018-08-27T15:45:59Z  generation: 1
            name: ocnssf-nfregistration  namespace: nssf-svc
            resourceVersion: "2336498"
            selfLink: /apis/extensions/v1beta1/namespaces/ nssf-svc/
deployments/ocnssf-nfregistration
            uid: 4b82fe89-aa10-11e8-95fd-fa163f20f9e2
            spec:  progressDeadlineSeconds: 600
            replicas: 1
            revisionHistoryLimit: 10
            selector:
            matchLabels:
            app: ocnssf-nfregistration
            strategy:
            rollingUpdate:
            maxSurge: 25%
            maxUnavailable: 25%
            type: RollingUpdate
            template:
            metadata:
            creationTimestamp: null
            labels:
            app: ocnssf-nfregistration
            spec:
            containers:
            - env:
            - name: MYSQL_HOST
            value: mysql
            - name: MYSQL_PORT
            value: "3306"
```

```
            - name: MYSQL_DATABASE
            value: nssfdb
            - name: nssf_REGISTRATION_ENDPOINT
            value: ocnssf-nfregistration
            - name: nssf_SUBSCRIPTION_ENDPOINT
            value: ocnssf-nfsubscription

            - name: NF_HEARTBEAT
            value: "120"
            - name: DISC_VALIDITY_PERIOD
            value: "3600"
            image: dsr-master0:5000/ocnssf-
nfregistration:latest
            imagePullPolicy: Always
            name: ocnssf-nfregistration
            ports:        - containerPort: 8080
            name: server
            protocol: TCP
            resources: {}
            terminationMessagePath: /dev/termination-log
            terminationMessagePolicy: File
            dnsPolicy: ClusterFirst
            restartPolicy: Always
            schedulerName: default-scheduler
            securityContext: {}
            terminationGracePeriodSeconds: 30
            status:
            availableReplicas: 1
            conditions:  - lastTransitionTime:
2018-08-27T15:46:01Z
            lastUpdateTime: 2018-08-27T15:46:01Z
            message: Deployment has minimum availability.
            reason: MinimumReplicasAvailable
            status: "True"
            type: Available
            - lastTransitionTime: 2018-08-27T15:45:59Z
            lastUpdateTime: 2018-08-27T15:46:01Z
            message: ReplicaSet
            "ocnssf-nfregistration-7898d657d9" has successfully
progressed.
            reason: NewReplicaSetAvailable
            status: "True"
            type: Progressing
            observedGeneration: 1
            readyReplicas: 1
            replicas: 1
            updatedReplicas: 1
```

- Check if the micro-services can access each other via REST interface. Execute following command:

  ```
  # kubectl -n <namespace> exec <pod name> -- curl <uri>
  ```

Example:

```
# kubectl -n nssf-svc exec $(kubectl -n nssf-svc get pods -o name|
cut -d'/' -f2|grep nfs) -
        curl http://ocnssf-nfregistration:8080/nnssf-nfm/v1/
nfinstances
# kubectl -n nssf-svc exec $(kubectl -n nssf-svc get pods -o name|
cut -d'/' -f2|grep nfr) -
curl http://ocnssf-nfsubscription:8080/nnssf-nfm/v1/nfinstances
```

> **Note:**
>
> These commands are in their simple form and display the logs only if there is 1 nssf<egistration> and nfs<ubscription> pod deployed.

- Application related tips
  Check the application logs and look for exceptions, by executing the following command:

  ```
  # kubectl -n <namespace> logs -f <pod name>
  ```

  You can use '-f' to follow the logs or 'grep' for specific pattern in the log output.

  Example:

  ```
  # kubectl -n nssf-svc logs -f $(kubectl -n nssf-svc get pods -o
  name|cut -d'/' -f2|grep nfr)
   # kubectl -n nssf-svc logs -f $(kubectl -n nssf-svc get pods -o
  name|cut -d'/' -f2|grep nfs)
  ```

  > **Note:**
  >
  > These commands are in their simple form and display the logs only if there is 1 nssf<egistration> and nfs<ubscription> pod deployed.

The environment is not working as expected

Problem: The environment is not working as expected.

Solution:

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

Problem: The user is getting the error: *failed to parse ocnssf-custom-values-1.4.0.yaml: error converting YAML to JSON: yaml.*

Solution:

Verify the following:

1. The `ocnssf-custom-values-1.4.0.yaml` may not be created properly.
2. The tree structure may not be followed.
3. There may be tab spaces in the file.
4. Verify that the `ocnssf-custom-values-1.4.0.yaml` is proper

   Refer *Network Slice Selection Function (NSSF) Cloud Native Installation Guide .*
5. If there is no error, helm installation will be deployed.

   Helm status can be checked using following command :

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

NSSF installation issues

Problem: The NSSF installation is not successful.

Solution:

1. Verify if NSSF specific pods are working as expected by executing the 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-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-client-nfdiscovery-cf48cd8d8-l4q2q     1/1  Running   0
   18m
   ocnssf-nrf-client-nfdiscovery-cf48cd8d8-vmt5v     1/1  Running   0
   18m
   ocnssf-nrf-client-nfmanagement-7db4598fbb-672hc   1/1  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
```

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-1.4.0.yaml`. Then, double check the image name and tags in `ocnssf-custom-values-1.4.0.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-1.4.0.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

Problem: The environment is not working as expected

Solution:

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

Collect the NSSF Logs to check the error scenarios

Problem: The error scenarios are checked by colllecting the NSSF logs.

Solution:

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
```

Custom Value File Parsing Failure

This section explains troubleshooting procedure in case of failure during parsing custom values file.

Problem

Not able to parse `ocnssf-custom-values-x.x.x.yaml`, while running helm install.

Error Code/Error Message Error:

failed to parse ocnssf-custom-values-x.x.x.yaml: error converting YAML to JSON: yaml

Symptom

While creating the `ocnssf-custom-values-x.x.x.yaml` file, if the above mentioned error is received, it means that the file is not created properly. The tree structure may not have been followed and/or there may also be tab spaces in the file.

Solution

Following the procedure as mentioned:

1. Download the latest NSSF templates zip file from OHC. Refer to Installation Tasks for more information.

2. Follow the steps mentioned in the Installation Tasks section.

Curl HTTP2 Not Supported

Problem

curl http2 is not supported on the system.

Error Code/Error Message

Unsupported protocol error is thrown or connection is established with HTTP/1.1 200 OK

Symptom

If unsupported protocol error is thrown or connection is established with http1.1, it is an indication that curl http2 support may not be present on your machine

Solution

Following is the procedure to install curl with HTTP2 support:

1. Make sure git is installed:

```
$ sudo yum install git -y
```

2. Install nghttp2:

```
$ git clone https://github.com/tatsuhiro-t/nghttp2.git
 $ cd nghttp2 $ autoreconf -i
$ automake
$ autoconf
$ ./configure
$ make
$ sudo make install
$ echo '/usr/local/lib' > /etc/ld.so.conf.d/custom-libs.conf
$ ldconfig
```

3. Install the latest Curl:

```
$ wget http://curl.haxx.se/download/curl-7.46.0.tar.bz2 (NOTE: Check
for latest version during Installation)
$ tar -xvjf curl-7.46.0.tar.bz2
$ cd curl-7.46.0
$ ./configure --with-nghttp2=/usr/local --with-ssl
$ make
$ sudo make install
$ sudo ldconfig
```

4. Make sure HTTP2 is added in features by executing the following command:

```
$ curl --http2-prior-knowledge -v "<http://10.75.204.35:32270/nnrf
disc/v1/nf-instances?requester-nf-type=AMF&target-nf-type=SMF>"
```

Tiller Pod Failure

Problem

Tiller Pod is not ready to run helm install.

Error Code/Error Message

The error 'could not find a ready tiller pod' message is received.

Symptom

When helm ls is executed, 'could not find a ready tiller pod' message is received.

Solution

Following is the procedure to install helm and tiller using the below commands:

1. Delete the pre-installed helm:

```
kubectl delete svc tiller-deploy -n kube-system kubectl delete deploy
tiller-deploy -n kube-system
```

2. Install helm and tiller using this commands:

```
helm init --client-only
helm plugin install https://github.com/rimusz/
helm-tiller helm tiller install
helm tiller start kube-system
```

Error Trigger based debugging

The following table lists the NSSF Alarms, cause for the errors and the solution:

Table 6-1 NSSF Events, causes and solution

Event	Type	Cause for the error	Solution
ocnssfPolicyNotFound	Alert	Rate of messages that did not find a matching policy is above warning threshold (Threshold: <>, Current: <>) Configuration issue: Operator did not configure Rules for the TAI + SNSSAI which is allowed as per AMF	Check TAI and SNSSAI for which there is no matching Rule. Add Rule to allocate slice for TAI+SNSSAI combination.
ocnssfNrfDiscFailed	Alert	NRF discovery message is getting error response	Check connectivity with NRF Validate NRF load
ocnssfNotificationFailure	Alert	Notification message towards AMF is failing	Validate connectivity to notification URL Check NS-Subscription load and capacity

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.

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 ocnssf-custom-values_1.4.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:


#############################################################
#            Section Start: global attributes           #
#############################################################

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

  # Kubernetes Secret containing DB credentials
  dbCredSecretName: 'ocnssf-db-creds'

  # NameSpace where secret is deployed
  nameSpace: ocnssf
```

```
# ********  Sub-Section Start: Ingress Gateway Global Parameters
********
#************************************************************************
***
 # 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

# ********  Sub-Section End: Ingress Gateway Global Parameters
********
#************************************************************************
***

# ********  Sub-Section Start: NRF CLIENT PARAMS Global Parameters
********
#************************************************************************
***

  # Jaeger tracing host
  envJaegerAgentHost: ''
  # Jaeger tracing port
  envJaegerAgentPort: 6831
  # Provide value for NodePort
  nrfClientNodePort: 0
  # Mysql Host Put NDB cluster IP in case MysQL pod is not being used
  envMysqlHost: ocnssf-nsdb.ocnssf
  # 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

# ********  Sub-Section End: NRF CLIENT  Global Parameters   ********
#************************************************************************
***

#############################################################
#            Section End  : global attributes           #
#############################################################




##############################################################################
###
#            Section Start: NSSF NSSelection Micro service attributes
#
##############################################################################
###
nsselection:
  image:
    # image name
    name: ocnssf-nsselection
    # image repository
    repository: reg-1:5000
    # tag name of image
    tag: 1.4.0
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # setting logging level
  # Possible values - OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE
  loglevel: "INFO"
  # MySql Host and Port configuration
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
  # NRF URL configuration
  nrf:
    primaryUrl: http://ocnrf.oracle.com:80
    secondaryUrl: http://ocnrf.oracle.com:80
# NSSF features flags
  features:
    nrfdiscovery: false # Flag to enable Discovery towards NRF to get
Candidate AMF set
    relevance: false # Flag to enable Relevence algorithm feature at
NSSF
    candidateResolution: true # Flag to enable Candidate AMF resolution
feature at NSSF
  # Flag to enable time based slice selection
  reqnftime: true
```

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


##############################################################################
###
#            Section End : NSSF NSSelection Micro service
attributes     #
##############################################################################
###

##############################################################################
###
#            Section Start: NSSF NSAvailability Micro service
attributes  #
##############################################################################
###
nsavailability:
  image:
    # image name
    name: ocnssf-nsavailability
    # image repository
    repository: reg-1:5000
    # tag name of image
    tag: 1.4.0
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # setting logging level
  # Possible values - OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE
  loglevel: "INFO"
  # MySql Host and Port configuration
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
  # Min replicas to scale to maintain an average CPU utilization
  minReplicas: 1
  # Max replicas to scale to maintain an average CPU utilization
  maxReplicas: 1

  #Expiry parameters for subscription
  maxExpiryDuration: 240 # range from 100 to 1000
  minExpiryDuration: 0 # range from 0 to 100

  # Enable/disable response gzip compression
  contentEncodingEnabled: true

  # Minimum response size required for compression to happen
```

```
  compressionMinimumResponseSize: 1024
##############################################################################
###
#            Section End : NSSF NSAvailability Micro service
attributes  #
##############################################################################
###

##############################################################################
###
#            Section Start: NSSF NSConfig Micro service attributes  #
##############################################################################
###
nsconfig:
  image:
    # image name
    name: ocnssf-nsconfig
    # image repository
    repository: reg-1:5000
    # tag name of image
    tag: 1.4.0
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # setting logging level
  # Possible values - OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE
  loglevel: "INFO"
  # MySql Host and Port configuration
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306

  # Min replicas to scale to maintain an average CPU utilization
  minReplicas: 1
  # Max replicas to scale to maintain an average CPU utilization
  maxReplicas: 1
  nrf:
    subscription: false # Flag to enable Subscriptions towards NRF for
AmfSet
  # URL at which NSSF receives notifications from Nrf. Set when NRF
subscription is turned ON.
  notificationHandlerUrl: http://ocnssf-ingress:80
##############################################################################
###
#            Section Start: NSSF NSConfig Micro service attributes  #
##############################################################################
###



##############################################################################
###
```

```
#            Section Start: NSSF NSSubscription Micro service
attributes  #
################################################################################
###

nssubscription:
  image:
    # image name
    name: ocnssf-nssubscription
    # image repository
    repository: reg-1:5000
    # tag name of image
    tag: 1.4.0
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # setting logging level
  # Possible values - OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE
  loglevel: "INFO"
  # MySql Host and Port configuration
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306

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

  # Maximum number of re trys towards AMF for notification , after
attempts notification is discarded
  httpMaxRetries: 2

  # 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

################################################################################
###
#            Section End : NSSF NSSubscription Micro service
attributes  #
################################################################################
###


################################################################################
###
#NSSF common micro services
#NrfClient : Used for interaction with NRF
#  nrf-client
```

```
#  app-info
#  perf-info
#  config-server
#Gateways : Used for ingress/egress gateway finctionalities,HTTPS
support ,OAuth Support and rate limiting
# Ingress-Gateway
# Egress-Gateway
##############################################################################
###

##############################################################################
###
#            Section Start: NrfClient Micro service attributes  #
##############################################################################
###
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":"310","mnc":"14"}],"fqdn":"nssf1.lab.oracle.com","interPlmnFqdn"
:"nssf1.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","v
ersions":
[{"apiVersionInUri":"v1","apiFullVersion":"1.15.3.0","expiry":"2019-12-3
1T23:59:59.000+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"
,"versions":
[{"apiVersionInUri":"v1","apiFullVersion":"1.15.3.0","expiry":"2019-12-3
1T23:59:59.000+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: nssfdb

  # 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

##############################################################################
###
#            Section End: NrfClient Micro service attributes  #
##############################################################################
###



##########################################################
#            Section Start: ingressgateway attributes   #
##########################################################
```

```
ingress-gateway:
  # This flag is for enabling/disabling HTTP/2.0  (insecure) in Ingress
Gateway.
  # If the value is set to false, NRF will not accept any HTTP/2.0
(unsecured) Traffic
  # If the value is set to true, NRF will accept HTTP/2.0 (unsecured)
Traffic
  enableIncomingHttp: true

  # This flag is for enabling/disabling HTTPS/2.0  (secured TLS) in
Ingress Gateway.
  # If the value is set to false, NRF will not accept any HTTPS/2.0
(secured) Traffic
  # If the value is set to true, NRF will accept HTTPS/2.0 (secured)
Traffic
  enableIncomingHttps: false

  # Ingress Gateway Service Container Image Details
  image:
    # Ingress Gateway image name
    name: ocingress_gateway
    # tag name of image
    tag: 1.7.3
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # Ingress Gateway Init  Container Image Details
  initContainersImage:
    # init Containers image name
    name: configurationinit
    # tag name of init Container image
    tag: 1.1.1
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # Ingress Gateway Update Container Image Details
  updateContainersImage:
    # update Containers image name
    name: configurationupdate
    # tag name of update Container image
    tag: 1.1.1
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # enable Jaeger tracing
  jaegerTracingEnabled: false
  openTracing :
    jaeger:
      udpSender:
        # Update this configuration when jaeger tracing is enabled.
        # udpsender host
        host: "jaeger-agent.cne-infra"
        # udpsender port
        port: 6831
      # Jaeger message sampler. Value range: 0 to 1
```

```
      # e.g. Value 0: No Trace will be sent to Jaeger collector
      # e.g. Value 0.3: 30% of message will be sampled and will be sent
to Jaeger collector
      # e.g. Value 1: 100% of message (i.e. all the messages) will be
sampled and will be sent to Jaeger collector
      probabilisticSampler: 0.5

  # Allowed CipherSuites for TLS1.2
  cipherSuites:
  - 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
  service:
   # configuration under ssl section is mandatory if
enableIncomingHttps is configured as "true"
    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:
```

```
    # setting logging level
    # Possible values - OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE
    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


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

##############################################################
#            Section Start: egressgateway attributes   #
##############################################################
egress-gateway:
```

```
egress-gateway:
  # This flag is for enabling/disabling HTTPS/2.0  (secured TLS) in
Egress Gateway.
  # If the value is set to false, NRF will send only HTTP/2.0
(unsecured) Egress Traffic
  # If the value is set to true, NRF will send only HTTPS/2.0 (secured)
Egress Traffic
  enableOutgoingHttps: false

  # Egress Gateway Service Container Image Details
  deploymentEgressGateway:
    # Egress Gateway image name
    image: ocegress_gateway
    # tag name of image
    imageTag: 1.7.3
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # Egress Gateway Init Container Image Details
  initContainersImage:
    # init Containers image name
    name: configurationinit
    # tag name of image
    tag: 1.1.1
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: IfNotPresent

  # Egress Gateway Update Container Image Details
  updateContainersImage:
    # update Containers image name
    name: configurationupdate
    # tag name of image
    tag: 1.1.1
    # Pull Policy - Possible Values are:- Always, IfNotPresent, Never
    pullPolicy: Always

  # enable Jaeger tracing
  jaegerTracingEnabled: false
  openTracing :
    jaeger:
      udpSender:
        # Update this configuration when jaeger tracing is enabled.
        # udpsender host
        host: "jaeger-agent.cne-infra"
        # udpsender port
        port: 6831
      # Jaeger message sampler. Value range: 0 to 1
      # e.g. Value 0: No Trace will be sent to Jaeger collector
      # e.g. Value 0.3: 30% of message will be sampled and will be sent
to Jaeger collector
      # e.g. Value 1: 100% of message (i.e. all the messages) will be
sampled and will be sent to Jaeger collector
      probabilisticSampler: 0.5

  # ********  Sub-Section Start: SCP released Parameters ********
```

```
  #*********************************************************************

  # Using SCP as an Proxy in Egress Gateway
  # If it is configured as false, SCP will not be used as an proxy.
  #   Messages will be directly sent to the Producers/HTTP Servers.
  # If it is configured as true, SCP will be used as an Proxy for
  #    delivering messages to the Producers/HTTP Servers.
  scpIntegrationEnabled: false

  # SCP Configuration For Egress Gateway
  # All the SCP related configuration will be used only
  # if scpIntegrationEnabled is set to true.
  #
  # SCP's HTTP Host/IP and Port Combination.
  # This will be while sending HTTP/2.0 (unsecured) traffic
  scpHttpHost: localhost
  scpHttpPort: 80

  # SCP's HTTPS Host/IP and Port Combination.
  # This will be while sending HTTPS/2.0 (secured) traffic
  scpHttpsHost: localhost
  scpHttpsPort: 443

  # SCP's API Prefix. (Applicable only for SCP with TLS enabled)
  # This will be used for constructing the Egress messgage's APIROOT
while proxying message to SCP.
  # Change this value to SCP's apiprefix. "/" is not expected to be
provided along.
  scpApiPrefix: /

  # SCP's default scheme  when 3gpp-sbi-target-apiroot header is missing
  scpDefaultScheme: https

  # ********  Sub-Section End : SCP released Parameters ********
  #*********************************************************************

  # Allowed CipherSuites for TLS1.2
  cipherSuites:
    - 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



  log:
    # setting logging level
    # Possible values - OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE
    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


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

 # ********  Sub-Section Start : OAUTH for notification Parameters
********
 #*********************************************************************

  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: ''
  # Flag to enable rate limiting for "notification" type of messages.
  notificationRateLimit:
    enabled: false
    duration: 60
    bucketCapacity: 4
    refillRate: 2
 # ********  Sub-Section end : OAUTH for notification Parameters
********
 #*********************************************************************


  #enable jagger tracing
  jaegerTracingEnabled: false

  openTracing:
    jaeger:
      udpSender:
        # udpsender host
        host: "occne-tracer-jaeger-agent.occne-infra"
        # udpsender port
        port: 6831
      probabilisticSampler: 0.5
#############################################################
#            Section End: egressgateway attributes   #
#############################################################
#############################################################
#            Section End: Common micro services   #
#############################################################
```