<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Acronyms</td>
<td>1-2</td>
</tr>
<tr>
<td>1-2</td>
<td>Admonishments</td>
<td>1-2</td>
</tr>
<tr>
<td>3-1</td>
<td>Installation Sequence</td>
<td>3-1</td>
</tr>
<tr>
<td>3-2</td>
<td>NSSF Images</td>
<td>3-1</td>
</tr>
<tr>
<td>4-1</td>
<td>NSSF Installation Preparation</td>
<td>4-1</td>
</tr>
<tr>
<td>5-1</td>
<td>Parameters and Definitions</td>
<td>5-1</td>
</tr>
<tr>
<td>5-2</td>
<td>NSSF Deployment</td>
<td>5-3</td>
</tr>
</tbody>
</table>
1

NSSF Overview

NSSF Overview

This section 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 set AMF to serve the UE. AMF can retrieve NRF, NSI ID, and target AMFs as part of UE initial registration and PDU establishment procedure.

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

The NSSF supports the following services:

**Nnssf_NSSelection Service**

This service is used by NF Service Consumer (AMF) to retrieve the information related to network slice. It also enables the NSSF to provide the AMF the Allowed NSSAI and the Configured NSSAI for the Serving PLMN.

**Nnssf_NSSelection Service**

This service supports GET request during the following procedures by UE:

- **Initial Register**: When the NSSF is able to detect authorized network slice information for the requested network slice selection information, the response body includes a payload body containing at least the allowed NSSAI, target AMF Set or the list of candidate AMF(s).

- **PDU session establishment**: This service allows to update the S-NSSAI(s), the NF service consumer such as AMF support on NSSF per TA basis, to subscribe and notify any change in status of the SNSSAIs available per TA (unrestricted) and the restricted S-NSSAI(s) per PLMN in that TA in the serving PLMN of the UE.

NSSF References

- CNE Installation Guide
- NSSF User's Guide

Acronyms

The following table provides information about the acronyms used in the document.
Table 1-1  Acronyms

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMF</td>
<td>Access and Mobility Management Function</td>
</tr>
<tr>
<td>NEF</td>
<td>Network Exposure Function</td>
</tr>
<tr>
<td>NF</td>
<td>Network Function</td>
</tr>
<tr>
<td>NRF</td>
<td>Network Repository Function</td>
</tr>
<tr>
<td>NSI ID</td>
<td>Network Slice Instance Identifier</td>
</tr>
<tr>
<td>NSSAI</td>
<td>Network Slice Selection Assistance Information</td>
</tr>
<tr>
<td>NSSF</td>
<td>Network Slice Selection Function</td>
</tr>
<tr>
<td>Network Slice</td>
<td>A logical network that provides specific network capabilities and</td>
</tr>
<tr>
<td></td>
<td>network characteristics</td>
</tr>
<tr>
<td>Network Slice instance</td>
<td>A set of Network Function instances and the required resources (e.g.</td>
</tr>
<tr>
<td></td>
<td>compute, storage and networking resources) which form a deployed</td>
</tr>
<tr>
<td></td>
<td>Network Slice</td>
</tr>
<tr>
<td>NF service</td>
<td>A functionality exposed by a NF through a service based interface</td>
</tr>
<tr>
<td></td>
<td>and consumed by other authorized NFs.</td>
</tr>
<tr>
<td>NSSP</td>
<td>Network Slice Selection Policy</td>
</tr>
<tr>
<td>PLMN</td>
<td>Public Land Mobile Network</td>
</tr>
<tr>
<td>Requested NSSAI</td>
<td>NSSAI provided by the UE to the Serving PLMN during</td>
</tr>
<tr>
<td></td>
<td>registration.</td>
</tr>
<tr>
<td>Allowed NSSAI</td>
<td>NSSAI provided by the Serving PLMN during e.g. a Registration</td>
</tr>
<tr>
<td></td>
<td>procedure, indicating the S-NSSAIs values the UE could use in the</td>
</tr>
<tr>
<td></td>
<td>Serving PLMN for the current registration area.</td>
</tr>
<tr>
<td>Configured NSSAI</td>
<td>NSSAI provisioned in the UE applicable to one or more PLMNs.</td>
</tr>
<tr>
<td>SEPP</td>
<td>Security Edge Protection Proxy</td>
</tr>
<tr>
<td>SST</td>
<td>Slice/Service type</td>
</tr>
<tr>
<td>SD</td>
<td>Slice Differentiator</td>
</tr>
<tr>
<td>S-NSSAI</td>
<td>Single Network Slice Selection Assistance Information</td>
</tr>
</tbody>
</table>

Documentation Admonishments

Admonishments are icons and text throughout this manual that alert the reader to assure personal safety, to minimize possible service interruptions, and to warn of the potential for equipment damage.

Table 1-2  Admonishments

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DANGER</td>
<td>Danger:</td>
</tr>
<tr>
<td></td>
<td>(This icon and text indicate the possibility of personal injury.)</td>
</tr>
<tr>
<td>WARNING</td>
<td>Warning:</td>
</tr>
<tr>
<td></td>
<td>(This icon and text indicate the possibility of equipment damage.)</td>
</tr>
</tbody>
</table>
Table 1-2 (Cont.) Admonishments

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caution:</td>
<td>(This icon and text indicate the possibility of service interruption.)</td>
</tr>
<tr>
<td>Topple:</td>
<td>(This icon and text indicate the possibility of personal injury and equipment damage.)</td>
</tr>
</tbody>
</table>

Locate Product Documentation on the Oracle Help Center Site

Oracle Communications customer documentation is available on the web at the Oracle Help Center site, http://docs.oracle.com. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at http://www.adobe.com.

2. Click Industries.
3. Under the Oracle Communications subheading, click Oracle Communications documentation link.
   The Communications Documentation page displays.
4. Click on your product and then the release number.
   A list of the documentation set for the selected product and release displays.
5. To download a file to your location, right-click the PDF link, select Save target as (or similar command based on your browser), and save to a local folder.

Customer Training

Oracle University offers training for service providers and enterprises. Visit our web site to view, and register for, Oracle Communications training:

http://education.oracle.com/communication

To obtain contact phone numbers for countries or regions, visit the Oracle University Education web site:

www.oracle.com/education/contacts

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.

**Emergency Response**

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or by calling the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system’s ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.
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:

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes</td>
<td>v1.12.5</td>
</tr>
<tr>
<td>HELM</td>
<td>v2.11.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Software</th>
<th>Chart Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>elasticsearch</td>
<td>1.21.1</td>
<td>Needed for Logging Area</td>
</tr>
<tr>
<td>elastic-curator</td>
<td>1.2.1</td>
<td>Needed for Logging Area</td>
</tr>
<tr>
<td>elastic-exporter</td>
<td>1.1.2</td>
<td>Needed for Logging Area</td>
</tr>
<tr>
<td>logs</td>
<td>2.0.7</td>
<td>Needed for Logging Area</td>
</tr>
<tr>
<td>kibana</td>
<td>1.5.2</td>
<td>Needed for Logging Area</td>
</tr>
<tr>
<td>grafana</td>
<td>2.2.0</td>
<td>Needed for Metrics Area</td>
</tr>
<tr>
<td>prometheus</td>
<td>8.8.0</td>
<td>Needed for Metrics Area</td>
</tr>
<tr>
<td>prometheus-node-exporter</td>
<td>1.3.0</td>
<td>Needed for Metrics Area</td>
</tr>
<tr>
<td>metallb</td>
<td>0.8.4</td>
<td>Needed for External IP</td>
</tr>
<tr>
<td>metrics-server</td>
<td>2.4.0</td>
<td>Needed for Metric Server</td>
</tr>
<tr>
<td>tracer</td>
<td>0.8.3</td>
<td>Needed for Tracing Area</td>
</tr>
</tbody>
</table>

**Note:**

In case any of the above services are needed and the respective software is not installed in CNE. Please install software before proceeding.

Network access

The Kubernetes cluster hosts 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 of the deployment. It may be some client machine (VM, server, local desktop so on).

Laptop/Desktop Client software

There are some requirements for the client machine where the deployment commands needs to be executed:

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

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

Table 3-1 Installation Sequence

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

NSSF Images

Following are the NSSF images:

Table 3-2 NSSF Images

<table>
<thead>
<tr>
<th>Services</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>ocnssf-nsconfig</td>
<td>ocnssf-nsconfig</td>
</tr>
<tr>
<td>ocnssf-nspolicy</td>
<td>ocnssf-nspolicy</td>
</tr>
<tr>
<td>ocnssf-nsregistration</td>
<td>ocnssf-nsregistration</td>
</tr>
<tr>
<td>ocnssf-nsseletion</td>
<td>ocnssf-nsseletion</td>
</tr>
</tbody>
</table>
# NSSF Installation Preparation

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

## Table 4-1  NSSF Installation Preparation

<table>
<thead>
<tr>
<th>Step</th>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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:<br>  
\[<\text{nfname}>-pkg-<\text{marketing-release-number}>\]
\[\text{For example: ocnssf-pkg-1.0.0.0.0.tgz}\] |
| 2    | Untar the NSSF Package File    | Untar the NSSF package to the specific repository:<br>  
\[\text{tar -xvf <<\text{nfname}>-pkg-<\text{marketing-release-number>>}\}]
\[\text{The package file consists of following:}\]
\[1. \text{ NSSF Docker Images File}\]
\[\text{ocnssf-images-1.0.0.tar}\]
\[2. \text{ Helm File}\]
\[\text{ocnssf-1.0.0.tgz}\]
\[3. \text{ Readme txt file}\]
\[\text{Readme.txt (Contains cksum and md5sum of tarballs)}\] |
| 3    | Check the checksums            | Check the checksums of tarballs mentioned in Readme.txt. Refer to the Readme.txt file for commands and checksum details.               |
| 4    | Load the tarball to system     | Execute the following command to push the Docker images to docker registry:<br>  
\[\text{docker load --input ocnssf-images-1.0.0.tar}\] |
| 5    | Check if all the images are    | Execute the following command to check:<br>  
\[\text{docker images}\]
\[\text{Refer table NSSF Images in section NSSF Installation Sequence for the list of images.}\] |
| 6    | Push docker images to docker   | Execute the following commands to push the docker images to docker registry:<br>  
\[\text{docker tag <image-name>:<image-tag> <docker-repo>/}\]
\[<image-name>:<image-tag>\]
\[\text{docker push <docker-repo>/<image-name>:<image-tag>}\] |
| 7    | Untar Helm Files               | Untar the helm files:<br>  
\[\text{tar -xvzf ocnssf-1.0.0.tgz}\] |
Table 4-1  (Cont.) NSSF Installation Preparation

<table>
<thead>
<tr>
<th>Step</th>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
</table>
| 8    | Download the Native Network Slice Selection Function (NSSF) Custom Template ZIP file | Download the **Native Network Slice Selection Function (NSSF) Custom Template** ZIP file from OHC:  
  - Go to the URL, docs.oracle.com  
  - Navigate to **Industries->Communications->Diameter Signaling Router->Cloud Native Network Elements**  
  - Click the **Native Network Slice Selection Function (NSSF) Custom Template** link to download the zip file.  
  - Unzip the template to get the following files:  
    - ocnssf-custom-values-1.0.0.yaml  
    - onssfDashboard.json |
NSSF Installation

This section includes information about NSSF deployment.

Following are the Parameters and Definitions used during NSSF deployment.

## Table 5-1 Parameters and Definitions

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;helm chart&gt;</td>
<td>It is the name of the chart that is of the form &lt;helm repo&gt;/ocnssf</td>
</tr>
<tr>
<td>&lt;OCNSSF version&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>&lt;release&gt;</td>
<td>It is a name provided by the user to identify the helm deployment</td>
</tr>
<tr>
<td>&lt;k8s namespace&gt;</td>
<td>It is a name provided by the user to identify the kubernetes namespace of the NSSF. All the NSSF micro services are deployed in this kubernetes namespace.</td>
</tr>
<tr>
<td>&lt;mysql host&gt;</td>
<td>It is the hostname of the mysql service and can be provided as, &lt;release&gt;-mysql.&lt;k8s namespace&gt;</td>
</tr>
</tbody>
</table>

### NSSF Deployment on Kubernetes

**Note:**

- To configure the parameters, see section NSSF Configuration.

### Create Database User/Group

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

The NSSF deployment using MySQL NDB cluster requires the database administrator to create user in 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 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 all the 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;

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

5. Create NSSF tables on one of the SQL Node:

```sql
create table nsi_profile (  
id integer not null AUTO_INCREMENT,  
name varchar(255) unique,  
nrf_uri varchar(255) not null,  
nsi_id varchar(255) unique,  
primary key (id)  
) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;

create table nsi_amfset_map (  
id integer not null AUTO_INCREMENT,  
nsi_profile_id integer not null,  
target_amf_region_id varchar(255) not null,  
target_amf_set_id varchar(255) not null,  
target_amf_set_fqdn varchar(255),  
salience integer not null,  
primary key (id),  
CONSTRAINT amf_validate UNIQUE (nsi_profile_id, target_amf_region_id, target_amf_set_id),  
foreign key (nsi_profile_id) references nsi_profile(id)  
) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;

create table nss_rule (  
id integer not null AUTO_INCREMENT,  
name varchar(255) unique not null,  
amf_id varchar(255),  
mcc varchar(50),  
mnc varchar(50),  
tac varchar(50),  
sst integer not null,  
sd varchar(50) not null,  
access_type varchar(50) not null,  
grant_field varchar(50) not null,  
salience integer not null,  
primary key (id),  
CONSTRAINT composite_key UNIQUE (amf_id, mcc, mnc, tac, sst, sd, access_type)  
) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;

create table nss_rule_profile_map (  
id integer not null AUTO_INCREMENT,  
rule_id integer not null,  
nsi_profile_id integer not null,  
salience integer not null,  
primary key (id),  
CONSTRAINT composite_key UNIQUE (rule_id, nsi_profile_id),  
foreign key (nsi_profile_id) references nsi_profile(id),  
foreign key (rule_id) references nss_rule(id)  
) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;
```
create table amf_resolution 
(
id integer not null AUTO_INCREMENT,
region_id varchar(255) not null,
set_id varchar(255) not null,
fqdn varchar(255),
instance_id varchar(255) not null,
primary key (id),
CONSTRAINT composite_key UNIQUE (region_id, set_id, instance_id)
) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;

create table configured_nssai 
(
id integer not null AUTO_INCREMENT,
amf_id varchar(255),
mcc varchar(50),
mnc varchar(50),
tac varchar(50),
salience integer not null,
primary key (id),
CONSTRAINT composite_key UNIQUE (amf_id, mcc, mnc, tac)
) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;

create table configured_nssai_map 
(
id integer not null AUTO_INCREMENT,
nssai_id integer not null,
sst integer not null,
sd varchar(50) not null,
primary key (id),
foreign key (nssai_id) references configured_nssai(id)
) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;

6. Exit from database and logout from SQL node.

Table 5-2  NSSF Deployment

<table>
<thead>
<tr>
<th>Step #</th>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | Create customized ocnssf-custom-values-1.0.0.yaml file | Create the customized ocnssf-custom-values-1.0.0.yaml file with the required input parameters.  
To configure the ocnssf-custom-values-1.0.0.yaml, see section NSSF Configuration or,  
The ocnssf-custom-values-1.0.0.yaml template can be downloaded from OHC.  
Download the package ocnssf-custom-configTemplates-1.0.0.0.0.zip and Unzip to get ocnssf-custom-values-1.0.0.yaml file. |
| 2      | Deploy OCNSSF | Execute the following command:  
```
helm install ocnssf/ --version <chart-version> --name <helm-release> --namespace <k8s namespace> -f ocnssf_customized_values.yaml
```
For example:  
```
helm install ocnssf/ --version 1.0.0 --name ocnssf --namespace ocnssf -f nssf-values-1.0.0.yaml
```
| 3      | Check status of the deployment | helm status <helm-release>  
For example: helm status ocnssf |
Table 5-2  (Cont.) NSSF Deployment

<table>
<thead>
<tr>
<th>Step #</th>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4      | Check status of the services | Execute the following command:  
kubectl get services -n <namespace>  
For example:  
kubectl get services -n ocnssf  
**Note:** If metallb is used, EXTERNAL-IP is assigned to ocnssf-nsgateway. |
| 5      | Check status of the pods | Execute the following command:  
kubectl get pods -n <ocnssf_namespace>  
Status column of all the pods should be ‘Running’  
Ready column of all the pods should be n/n, where n is number of containers in the pod.  
For example:  
kubectl get pods -n ocnssf |

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ocnssf-nsconfig-6c498d9f99-82xnm</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>ocnssf-nsgateway-5479cc694d-fv5zr</td>
<td>2/2</td>
<td>Running</td>
</tr>
<tr>
<td>ocnssf-nsgateway-5479cc694d-kvplh</td>
<td>2/2</td>
<td>Running</td>
</tr>
<tr>
<td>ocnssf-nspolicy-6df8dff468-8tfn7</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>ocnssf-nspolicy-6df8dff468-phwz8</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>ocnssf-nsregistration-67c54fd586-vhcqp</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>ocnssf-nsselection-6756447685-6b9ms</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>ocnssf-nsselection-6756447685-vvxbs</td>
<td>1/1</td>
<td>Running</td>
</tr>
</tbody>
</table>
6

NSSF Uninstallation

Deleting the NSSF deployment

To completely delete or remove the NSSF deployment, execute:

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

For example:

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

Delete kubernetes namespace

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

For example:

```bash
kubectl delete namespace ocnssf
```
NSSF Configuration

This section includes information about applicable configuration supported during NSSF deployment.

NSSF supports the following configuration:

**Note:**

Perform the provided configurations, once docker platform is configured for NSSF. Check if the Registry folders are in place having the latest helm charts and jar files for NSSF node.

Prepare `values.yaml` file and update configuration parameters for each micro service. Refer `sample_values.yaml` for further details.

**Nsselection**

- Set the appropriate logging level:
  
  Provide value for logging level (one of: OFF, INFO, DEBUG, ERROR, ALL)

  `nsselection.nss.loglevel`

**Nsconfig**

- Configure SQL DB cluster
  
  - Set the hostname of the primary sql node in `nsconfig.mysql.primary.host`
  
  - Set the hostname of the secondary sql node in `nsconfig.mysql.secondary.host`

  - Set the appropriate logging level
    
    Provide value for logging level (one of: OFF, INFO, DEBUG, ERROR, ALL)

    `nsconfig.nsc.loglevel`
Nsregistration

- Configure the NSSF profile
  - Set NSSF Profile parameters for registration with NRF at `nsregistration.configFiles.nssf.profile`

- Configure PLMN level NRF route:
  - Set the fqdn and port of PLMN level NRF in `nsregistration.nrf.host` and `nsregistration.nrf.port`

- Configure the NSSF fqdn
  - Provide value of FQDN of NSSF node at `nsregistration.nssf.fqdn`

- Set the appropriate logging level
  - Provide value for logging level (one of: OFF, INFO, DEBUG, ERROR, ALL) `nsconfig.nsc.loglevel`
Sample values.yaml file

This section includes a sample of the values.yaml file.

The following sample illustrates the values.yaml file:

# Copyright 2019 (C), Oracle and/or its affiliates. All rights reserved.
#
# This yaml file could be supplied in helm install command when deploying OCNSSF
# v1.x.y
#
# e.g. helm install <helm-repo>/ocnssf --name ocnssf --namespace ocnssf -f <this
# file>
#
# Compatible with OCNSSF CHART VERSION 1.x.y
#
# - Please set the docker registry (repository) in
#   nsselection.image.repository
#   nspolicy.image.repository
#   nsconfig.image.repository
#   nsregistration.image.repository
#
# - To integrate with NDB DB-Tier cluster,
#   set the hostname of the primary sql node in:
#   nspolicy.mysql.primary.host
#   nsconfig.mysql.primary.host
#   AND
#   set the hostname of the secondary sql node in
#   nspolicy.mysql.secondary.host
#   nsconfig.mysql.secondary.host
#   AND
#   set the username and password in
#   nspconfig.mysql.username
#   nsconfig.mysql.password
#   (note: the username and password can be provided in an alternative way by
#    creating a kubernetes secret
#   with the following properties:
#   name: ocnssf-creds
#   namespace: same as ocnssf deployment
#   data:
#      mysql-username: <username>
#      mysql-password: <password>
#
# - To connect to the PLMN level NRF to utilize registration services
#   set the fqdn and port of NRF in:
#   nsregistration.nrf.host and nsregistration.nrf.port
#   AND
#   configure the NSSF profile in
#   nsregistration.configFiles.nssf.profile
#   AND
#   set the NSSF fqdn in
#   nsregistration.nssf.fqdn
#
# - To connect to the PLMN level NRF to utilize discovery service for AMF candidates
#   set the fqdn and port of NRF in:
#   nsPolicy.nrf.host and nsPolicy.nrf.port
#   AND
#   set nsPolicy.nsp.nrfdisc to true
#
# - To turn on logging set the appropriate logging level (one of: OFF, INFO, DEBUG, ERROR, ALL)
# in one or more of the following:
#   nsSelection.nss.loglevel
#   npolicy.nsp.loglevel
#   nsRegistration.nsr.loglevel
#   nsConfig.nsc.loglevel

nsSelection:
  image:
    repository: reg-1:5000
  nss:
    loglevel: "OFF"

nsPolicy:
  image:
    repository: reg-1:5000
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
  nrf:
    host: http://ocnrf.oracle.com
    port: 80
  nsp:
    loglevel: "OFF"
    nrfdisc: true

nsConfig:
  image:
    repository: reg-1:5000
  mysql:
    primary:
      host: "ocnssf-nsdb.ocnssf"
    secondary:
      host: "ocnssf-nsdb.ocnssf"
    port: 3306
    username: changeme
    password: changeme
  nsc:
    loglevel: "OFF"

nsRegistration:
  configFiles:
    nssf.profile: |
    {
      "plmn": {"mcc": "310", "mnc": "14"},
      "fqdn": "ocnssf-nsgateway.ocnssf.svc.us.lab.oracle.com",
      "interpPlmnFqdn": "ocnssf-nsgateway.ocnssf.svc.us.lab.oracle.com",
      "ipv4Addresses": [ "127.0.0.1", "10.0.0.1" ],
      "ipv6Addresses": [ "::1", "::2" ],
"priority": 5,
"locality": "us-east",
"nfServices": [{
  "serviceName": "nssf-nsselection",
  "versions": [{
    "apiVersionInUri": "v1",
    "apiFullVersion": "1.R15.2.0",
    "expiry": "2019-12-31T23:59:59.000+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
  }]
}]
image:
  repository: reg-1:5000
nrf:
  host: http://ocnrf.oracle.com
  port: 80
nssf:
  fqdn: ocnssf-nsgateway.ocnssf.svc.us.lab.oracle.com
nsr:
  loglevel: "OFF"

nsgateway:
  service:
    annotations:
      metallb.universe.tf/address-pool: signaling
    adminService:
      create: false