## Contents

1. **Introduction**
   - References 1-1
   - Acronyms 1-1
   - Locate Product Documentation on the Oracle Help Center Site 1-2
   - Documentation Admonishments 1-2
   - Customer Training 1-3
   - My Oracle Support 1-3
   - Emergency Response 1-4

2. **Installing Network Exposure Function**
   - Pre-requisites 2-1
   - Installation Sequence 2-2
   - Creating Database Account on MySQL Database 2-2
   - Installation Preparation 2-5
   - Deploying Network Exposure Function 2-6
   - Verifying Installation 2-9

3. **Configuring Network Exposure Function**
   - Get nrf Service URL 3-1
   - Get NEF Profile 3-1
   - Updating nrf Client Service Related configmap 3-1
   - Verifying kong API Gateway Service 3-2
     - Known Behavior About Failed kong-migration Pod 3-7
   - Enabling Loadbalancer with MetalLB 3-7
     - Updating API-Gateway Service 3-7

4. **Upgrading Network Exposure Function**
   - Verifying Upgrade 4-2
5  Uninstalling Network Exposure Function

Verifying Uninstallation
List of Figures

3-1 Kong-migration Pod 3-7
# List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Acronyms</td>
<td>1-1</td>
</tr>
<tr>
<td>1-2</td>
<td>Admonishments</td>
<td>1-3</td>
</tr>
<tr>
<td>2-1</td>
<td>Docker Images</td>
<td>2-6</td>
</tr>
<tr>
<td>2-2</td>
<td>Module Descriptions</td>
<td>2-7</td>
</tr>
<tr>
<td>2-3</td>
<td>Variables Description</td>
<td>2-8</td>
</tr>
<tr>
<td>2-4</td>
<td>Service Deployment Service Type</td>
<td>2-9</td>
</tr>
<tr>
<td>3-1</td>
<td>Variables</td>
<td>3-7</td>
</tr>
<tr>
<td>4-1</td>
<td>Parameters for Deployment</td>
<td>4-2</td>
</tr>
</tbody>
</table>
1

Introduction

The Network Exposure Function (NEF) is a functional element that supports the following functionalities:

- Securely exposes network capabilities and events provided by 3GPP Network Functions to AF.
- Provides a means for the AF to securely provide information to 3GPP network and may authenticate, authorize, and assist in throttling the AF.
- Translates the information received from the AF to the one sent to internal 3GPP NFs, and vice versa.
- Supports to expose information (collected from other 3GPP NFs) to the AF.
- Supports a PFD function that allows the AF to provision PFD(s) and may store and retrieve PFD(s) in the UDR. The NEF further provisions PFD(s) to the SMF.

A specific NEF instance may support one or more of the functionalities described above and consequently an individual NEF may support a subset of the APIs specified for capability exposure.

References

Refer to the following documents for more information about 5G cloud native network exposure function.

- CNE Installation Document

Acronyms

The following table provides information about the acronyms used in the document.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5GC</td>
<td>5G Core Network</td>
</tr>
<tr>
<td>5GS</td>
<td>5G System</td>
</tr>
<tr>
<td>AF</td>
<td>Application Function</td>
</tr>
<tr>
<td>AUSF</td>
<td>Authentication Server Function</td>
</tr>
<tr>
<td>BSF</td>
<td>Binding Support Function</td>
</tr>
<tr>
<td>CHF</td>
<td>Charging Function</td>
</tr>
<tr>
<td>CNE</td>
<td>Cloud Native Environment</td>
</tr>
<tr>
<td>GPSI</td>
<td>Generic Public Subscription Identifier</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>
</tbody>
</table>
Table 1-1  (Cont.) Acronyms

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSSF</td>
<td>Network Slice Selection Function</td>
</tr>
<tr>
<td>PCF</td>
<td>Policy Control Function</td>
</tr>
<tr>
<td>PFD</td>
<td>Packet Flow Description</td>
</tr>
<tr>
<td>QFI</td>
<td>QoS Flow Identifier</td>
</tr>
<tr>
<td>QoE</td>
<td>Quality of Experience</td>
</tr>
<tr>
<td>SEPP</td>
<td>Security Edge Protection Proxy</td>
</tr>
<tr>
<td>SBA</td>
<td>Service Based Architecture</td>
</tr>
<tr>
<td>SMF</td>
<td>Session Management Function</td>
</tr>
<tr>
<td>SUPI</td>
<td>Subscription Permanent Identifier</td>
</tr>
<tr>
<td>UDR</td>
<td>Unified Data Repository</td>
</tr>
<tr>
<td>UDSF</td>
<td>Unstructured Data Storage Function</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 (OHC) site, [http://docs.oracle.com](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](http://www.adobe.com).

1. Access the Oracle Help Center site at [http://docs.oracle.com](http://docs.oracle.com).
2. Click Industries.
3. Under the Oracle Communications subheading, click the Oracle Communications documentation link.

   The Communications Documentation page appears. Most products covered by these documentation sets will appear under the headings "Network Session Delivery and Control Infrastructure" or "Platforms."

4. Click on your Product and then the Release Number.
   A list of the entire documentation set for the selected product and release appears.

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.

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: (This icon and text indicate the possibility of personal injury.)</td>
</tr>
<tr>
<td>![WARNING]</td>
<td>Warning: (This icon and text indicate the possibility of equipment damage.)</td>
</tr>
<tr>
<td>![CAUTION]</td>
<td>Caution: (This icon and text indicate the possibility of service interruption.)</td>
</tr>
</tbody>
</table>

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

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

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select 2 for New Service Request.
2. Select 3 for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
   • For Technical issues such as creating a new Service Request (SR), select 1.
   • For Non-technical issues such as registration or assistance with My Oracle Support, select 2.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.
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.
Installing Network Exposure Function

This section provides instructions for installing Network Exposure Function.

Pre-requisites

Following are pre-requisites user must have before installing Network Exposure Function.

<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>
<tr>
<td>MySQL</td>
<td>5.7 or later</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-curater</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 Logging Area</td>
</tr>
<tr>
<td>prometheus</td>
<td>8.8.0</td>
<td>Needed for Logging 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 Metrics 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.

Note:

If you are using NRF, install it before proceeding with the NEF installation.

Network Access

The Kubernetes cluster hosts must have network access to:
- quay.io/datawire/ambassador docker image repository
- Local helm repository where the Oracle Communications Network Exposure Function helm charts are available. The helm charts are available in tar ball with images.
- Local docker image repository where the Oracle Communications Network Exposure Function images are available. The tar ball has the images which are posted to your repository.

**Laptop/Desktop Client Software**

Following are the requirements for the laptop/desktop where the deployment commands shall be executed:

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

**Browser Support**

It is recommended to use Firefox browser to access Kubernetes dashboard. The Configuration Management GUI page is accessed from different browsers.

**Server or space requirements**

For server and space requirements, refer to *Oracle Communications Cloud Native Environment Installation Guide*.

### Installation Sequence

This section provides the order in which you shall perform the NEF installation.

1. Create a MySQL database account. See *Creating Database Account on MySQL Database*
2. Download NEF package files and load them to the system. See *Installation Preparation*.
3. Prepare all variables for Helm install command. See Table 2-1
4. NEF Deployment using Helm command. See *Deploying Network Exposure Function*
5. Verify NEF Deployment. See *Verifying Installation*.
6. Configure NEF. See *Configuring Network Exposure Function*.

### Creating Database Account on MySQL Database

Create a new database account on MySQL database by executing the following command:

```
CREATE USER 'nefusr'@'%' IDENTIFIED BY 'nefpasswd';
GRANT ALL PRIVILEGES ON *.* TO 'nefusr'@'%';
```

Login to MySQL console as a new user created above:
mysql -h<MYSQL_HOST> -u<USERNAME> -p<PASSWORD>

Execute the below script to initial Network Exposure Function databases. At first login to MySQL console via new user created above,

```sql
mysql -h<MYSQL_HOST> -unefusr -pnefpasswd

CREATE DATABASE IF NOT EXISTS `ocpm_config_server`;

CREATE TABLE IF NOT EXISTS `ocpm_config_server`.`topic_info` (  
    `id` bigint(20) NOT NULL AUTO_INCREMENT,  
    `description` varchar(255) COLLATE utf8_unicode_ci DEFAULT 'Default Topics.',  
    `name` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
    `modify_date` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    `version` int(11) NOT NULL,  
    PRIMARY KEY (`id`),  
    UNIQUE KEY `UK_gd6b0a6mdpxc55qbibre2cldc` (`name`)  
) AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

CREATE TABLE IF NOT EXISTS `ocpm_config_server`.`configuration_item` (  
    `id` bigint(20) NOT NULL AUTO_INCREMENT,  
    `cfg_key` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
    `md5sum` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
    `cfg_value` mediumtext COLLATE utf8_unicode_ci,  
    `version` int(11) NOT NULL,  
    `topic_info_id` bigint(20) NOT NULL,  
    PRIMARY KEY (`id`),  
    KEY `FKdue8drxn6acrdt63iacireky1` (`topic_info_id`)  
) DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

insert into `ocpm_config_server`.`topic_info` (name, version) values ('policy', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('policySchema', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('policyElement', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('policyParam', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('policygui', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.global.cfg', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.smservice.cfg', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.public.sessionrule', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.public.sessionruleprofile', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.public.authorizeddefaultqos', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.public.pccrule', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.public.qosdata', 1);
insert into `ocpm_config_server`.`topic_info` (name, version) values ('pcf.public.chargingdata', 1);
```
insert into `ocpm_config_server`.'topic_info' (name, version) values ('pcf.public.pccruleprofile', 1);

insert into `ocpm_config_server`.'topic_info' (name, version) values ('amservice.system', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('public.matchlist', 1);

insert into `ocpm_config_server`.'topic_info' (name, version) values ('pe.serviceTag', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('pe.policyTag', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('pe.logLevel', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('pcf.amservice.app', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('nrfclient.cfg', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('NRF.UDR', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('NRF.BSF', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('pcf.userservice.cfg', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('NRF.CHF', 1);
insert into `ocpm_config_server`.'topic_info' (name, version) values ('pcf.chfservice', 1);

CREATE DATABASE IF NOT EXISTS `ocpm_nef_me`;
CREATE TABLE IF NOT EXISTS `ocpm_nef_me`.'mesubscription' (  
`id` int(11) NOT NULL,  
`me_subscription` json DEFAULT NULL,  
`tltrid` varchar(255) DEFAULT NULL,  
PRIMARY KEY (`id`)  
) DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

CREATE TABLE IF NOT EXISTS `ocpm_nef_me`.'hibernate_sequence' (  
`next_val` bigint(20) DEFAULT NULL  
) DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

CREATE DATABASE IF NOT EXISTS `ocpm_capif_core`;
CREATE TABLE IF NOT EXISTS `ocpm_capif_core`.'aef_api_profile' (  
`apf_id` varchar(255) NOT NULL,  
`api_id` varchar(255) NOT NULL,  
`aef_profiles` json DEFAULT NULL,  
`api_name` varchar(255) DEFAULT NULL,  
`description` varchar(255) DEFAULT NULL,  
`supported_features` varchar(255) DEFAULT NULL,  
PRIMARY KEY (`apf_id`,`api_id`)  
) DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

CREATE TABLE IF NOT EXISTS `ocpm_capif_core`.'invoker_profile' (  
`invoker_id` varchar(255) NOT NULL,  
`api_invoker_certificate` varchar(255) DEFAULT NULL,  
`api_invoker_public_key` varchar(255) DEFAULT NULL,  
`notification_destination` varchar(255) DEFAULT NULL,  
`onboarding_secret` varchar(255) DEFAULT NULL,  
`request_test_notification` bit(1) DEFAULT NULL,  
`request_websocket_uri` bit(1) DEFAULT NULL,
Installation Preparation

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

1. Download the NEF package file from Oracle Software Delivery Cloud (OSDC) at: http://edelivery.oracle.com. Package is named as follows:
   `<nfname>-pkg-<marketing-release-number>.tgz`
   
   For example, `ocnef-pkg-1.0.0.0.tgz`

2. Untar the NEF Package File.
   `tar -xvf <<nfname>-pkg-<marketing-release-number>>.tgz`
   
   This command results into `<<nfname>-pkg-<marketing-release-number>>` directory.
   
   The directory consists of following:

   - **NEF Docker Images File**: `ocnef-images-1.0.0.tar`
   - **Helm File**: `ocnef-1.0.0.tgz`
   - **Readme txt File**: `Readme.txt` (Contains cksum and md5sum of tarballs)

3. Verify the checksums of tarballs mentioned in `Readme.txt`. 
Deploying Network Exposure Function

**Note:**
The Network Exposure Function requires a MySQL database to store the configuration and run time data.

To deploy the NEF:

1. Download the file, `ocnef-pkg-1.0.0.0.0.tgz`.
2. Untar `ocnef-pkg-1.0.0.0.0.tgz`.
3. Untar displays the following files:
   - `ocnef-1.0.0.tgz` (helm chart)
   - `ocnef-images-1.0.0.tar` (docker images)
   - `Readme.txt` (Contains checksum and md5sum of tarballs)
4. Check the checksums of tarballs mentioned in the `Readme.txt` file.
5. After you load the tarballs to docker images, if required, re-tag it according to your specific repository.
   - Run the following command to load `ocnef-images-1.0.0.tar` to docker and push imported docker images to user docker registry.
     
     ```
     docker load --input /<IMAGE_PATH>/ocnef-images-1.0.0.tar
     docker tag ocpcf/nef_asqos:1.0.0 <customer repo>/nef_asqos:1.0.0
     docker push <customer repo>/nef_asqos:1.0.0
     ```

   * Repeat above tag and push commands for ALL imported docker images as listed in the Table 2-1

**Note:**
User may need to configure docker certificate to access customer registry via HTTPS. Configure the certificate before executing the docker push command or the command may fail to execute.

Table 2-1 provides the details of the docker images file name:

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Docker Image Name</th>
<th>Service Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Service</td>
<td>db-service</td>
<td>Common</td>
</tr>
<tr>
<td>Nrf Client Service</td>
<td>nrf_clientservice</td>
<td>Common</td>
</tr>
<tr>
<td>CM Service</td>
<td>ocpm_cm_service</td>
<td>Common</td>
</tr>
<tr>
<td>Performance Monitoring</td>
<td>perf_info</td>
<td>Platform</td>
</tr>
</tbody>
</table>
### Table 2-1 (Cont.) Docker Images

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Docker Image Name</th>
<th>Service Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config Server Service</td>
<td>ocpm_config_server</td>
<td>Common</td>
</tr>
<tr>
<td>Asqos Service</td>
<td>nef_asqos</td>
<td>NEF</td>
</tr>
<tr>
<td>Nef Me Service</td>
<td>nef_me</td>
<td>NEF</td>
</tr>
<tr>
<td>Capif Core Service</td>
<td>capif_core</td>
<td>NEF</td>
</tr>
<tr>
<td>Application Info Service</td>
<td>app_info</td>
<td>Platform</td>
</tr>
<tr>
<td>Readiness Check</td>
<td>readiness-detector</td>
<td>Common</td>
</tr>
<tr>
<td>Oracle Linux 7 with JDK11</td>
<td>ocpm_ol7_jdk11</td>
<td>Common</td>
</tr>
</tbody>
</table>

Table 2-2 provides the information about the modules:

### Table 2-2 Module Descriptions

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>common</td>
<td>Common service module which would be shared by separate NF as supporting service, such as GUI, and API Gateway.</td>
</tr>
<tr>
<td>platform</td>
<td>Platform service to provide monitoring services, also shared by separate NF.</td>
</tr>
<tr>
<td>nef</td>
<td>Define NEF services.</td>
</tr>
</tbody>
</table>

6. Execute the following command:

```bash
helm install --namespace=<NAMESPACE>-nef --name=<NAME>-nef \
  --set global.envMysqlHost=<MYSQL_HOST>,global.envMysqlUser=nefusr,global.envMysqlPassword=nefpasswd \
  --set global.envJaegerAgentHost=<JAEGER_SERVICE>.<JAEGER_SERVICE_NAMESPACE> \
  --set global.envManageNF=NEF,global.envSystemName=NEF,common.configmapApplicationConfig.nrfClientType=NEF \
  --set global.imageTag=<IMAGE_TAG>,global.dockerRegistry=<DOCKER_REGISTRY_ADDRESS> \
  --set platform.enabled=true,pcf.enabled=false,bsf.enabled=false,common.enabled=true,nef.enabled=true \
  --set common.deploymentNrfClientService.envNamespace=<NAMESPACE>-nef,platform.appinfo.topic=NRF.UDM
```

**Note:**

It is mandatory to run the below command under helm chart folder as the last line of the command, `./<HELM_CHART_NAME_WITH_EXTENSION>` specifies that helm chart path is current working path. To run the below command in another server, copy the helm chart file to it first.
Table 2-3 provides details of each variable:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NAMESPACE&gt;</td>
<td>Deployment NF namespace used by helm command</td>
<td>global.envMysqlUser and global.envMysqlPassword variables in above command from database section configured in previous step</td>
</tr>
<tr>
<td>&lt;NAME&gt;</td>
<td>Deployment NF name used by helm command</td>
<td>global.envMysqlUser and global.envMysqlPassword variables in above command from database section configured in previous step</td>
</tr>
<tr>
<td>&lt;MYSQL_HOST&gt;</td>
<td>MySQL host name or IP address</td>
<td>global.envMysqlUser and global.envMysqlPassword variables in above command from database section configured in previous step</td>
</tr>
<tr>
<td>&lt;JAEGGER_SERVICE&gt;</td>
<td>Both parameters could be found in same Kubernetes cluster</td>
<td>Follow the below format:</td>
</tr>
<tr>
<td>&lt;JAEGGER_SERVICE_NAMESPACE&gt;</td>
<td>Both parameters could be found in same Kubernetes cluster</td>
<td>Follow the below format:</td>
</tr>
<tr>
<td></td>
<td>For example, consider OCNNE. Run the command, kubectl get svc -n occne-infra</td>
<td>For example, consider OCNNE. Run the command, kubectl get svc -n occne-infra</td>
</tr>
<tr>
<td></td>
<td>Consider *-jaeger-agent as jaeger service name, such as occne-tracer-jaegeragent.occne-infra</td>
<td>Consider *-jaeger-agent as jaeger service name, such as occne-tracer-jaegeragent.occne-infra</td>
</tr>
<tr>
<td></td>
<td>Example: occne-tracer-jaegeragent.occne-infra</td>
<td>Example: occne-tracer-jaegeragent.occne-infra</td>
</tr>
<tr>
<td>&lt;IMAGE_TAG&gt;</td>
<td>The image tag used in customer docker registry, it is recommend to use same image tag when pull docker image to registry. If follow above steps to push docker image to customer docker registry then the &lt;IMAGE_TAG&gt; value should be 1.0.0</td>
<td>Each service deployment yaml file would use global.imageTag as image tag to fetch related docker image per helm chart design. With the release tar file, the global image tag for all services is 1.0.0</td>
</tr>
<tr>
<td>&lt;DOCKER_REGISTRY_ADDRESS&gt;</td>
<td>Customer docker registry address</td>
<td>If registry has port value, add port. For example, reg-1:5000</td>
</tr>
</tbody>
</table>

NEF Services deployment service type. Kubernetes provides the following three deployment types.
Table 2-4  Service Deployment Service Type

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClusterIP</td>
<td>Exposes the service on a cluster-internal IP. Choosing this value makes the service only reachable from within the cluster.</td>
</tr>
<tr>
<td></td>
<td>This is the default ServiceType</td>
</tr>
<tr>
<td>NodePort</td>
<td>Exposes the service on each Node's IP at a static port (the NodePort). A ClusterIP service, to which the NodePort service routes is automatically created. Contact the NodePort service from outside the cluster, by requesting, <code>&lt;NodeIP&gt;:&lt;NodePort&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Most of the NEF services use NodePort to deploy.</td>
</tr>
<tr>
<td>LoadBalancer</td>
<td>Exposes the service externally using a cloud provider's load balancer. NodePort and ClusterIP services, to which the external load balancer will route, are automatically created. Given latest OCCNE already integrated METALLB, configure IP address to MetalLB on OCCNE.</td>
</tr>
</tbody>
</table>

Note:

For user interface page and API gateway service, it is mandatory to use loadBalancer type. Assuming that OCCNE is integrated with MetalLB, configure IP address to MetalLB on OCCNE.

Verifying Installation

To verify installation, run the following command:

```
kubectl get svc -n <NEF-Namespace>
kubectl get pod -n <NEF-Namespace>
```

If installation is successful, then all pods display in Running/Completed status except there are some error pods named kong-migration per Known Behavior About Failed kong-migration Pod.
3

Configuring Network Exposure Function

Configuring Network Exposure Function involves the following:

- Updating nrf Client Service Related configmap
- Verifying kong API Gateway Service

Get nrf Service URL

Run the following command and get nrf service name (if you have installed Oracle NRF, then service name is `ocnrf-endpoint`) and service port (default port value is 80).

```
kubectl get svc -n <NRF-Namespace>
```

For example, `<NRF-Namespace>` variable is `nrf1-1` and nrf service name `ocnrf-endpoint`, then the nrf service value is `ocnrf-endpoint.nrf1-1:80`.

Get NEF Profile

The NEF Profile here indicates the entrance of current deployed NEF service sets, which means it is the API Gateway service URL (FQDN), API Gateway service IP V4 address or IP V6 address.

Get API Gateway Service FQDN via kubectl Command

Run the following command and get API gateway service name, `<NEF-Name>-api-gateway` with default application port value 80.

```
kubectl get svc -n <NEF-Namespace>
```

For example, consider `<NEF-Namespace>` variable, as `nef1-1` and API gateway service name as `ocnef-api-gateway`, then the API gateway service value is `ocnef-api-gateway.nef1:80`.

Updating nrf Client Service Related configmap

User may have to update nrf client service related configmap. Following are few scenarios:

- Currently deployed NEF uses updated configmap to register NEF itself to NRF service.
- Client gets NEF related FQDN, IPv4, or IPv6 address from NRF service and send request to specific service under NEF with that FQDN, IPv4 or IPv6 address. The request reaches to API Gateway service under NEF deployment.
- The API Gateway service under NEF would parse coming request, then distributes that request to specific service per request URL and forward response from upstream service to client.

The nrf client configmap establish connection between NRF and NEF, in order to register NEF to NRF services and hence it is mandatory to update configmap to use correct nrf service URL and apply some extra settings.
Before updating nrf client service, perform the following:

- Get NRF Service URL
- Get NEF Profile

To update the nrf client service config map:

1. Run the following command and get config map item, `<NEF-NAME>-application-config`:
   ```bash
kubectl get configmap -n <NEF-Namespace>
```
2. Run the following command and then navigate to vim editor with config map content displayed
   ```bash
cubectl edit configmap <NEF-NAME>-application-config -n <NEF-Namespace>
```
3. Update `nrfApiRoot` to actual deployed nrf service URL. For example, `nrfApiRoot=http://ocnr1-endpoint.nrf1-1:80`. See Get NRF Service URL to get actual nrf service URL.

At least one of the address parameters such as fqdn, ipv4address or ipv6address should be populated in the NF Profile under configmap definition. NF Profile here indicates the entrance of current deployed NEF which means it is the API Gateway service URL (FQDN), API Gateway service IP V4 address or IP V6 address.

```json
appProfiles=[{"nfInstanceId": "fe7d882b-0541-4c7d-ab84-c6d70b1b0123","nfType": "PCF","nfStatus": "REGISTERED","plmn": null,"nsiList": null,"fqdn": null,"interPlmnFqdn": null,"ipv4Addresses": null,"ipv6Addresses": null ........
```

It is recommended to update the fqdn value while keeping ipv4Addresses and ipv6Addresses as it is.

For example, consider ocnf-api-gateway.nef1-1:80 for fqdn value. Following result displays:

```json
appProfiles=[{"nfInstanceId": "fe7d882b-0541-4c7d-ab84-c6d70b1b0123","nfType": "PCF","nfStatus": "REGISTERED","plmn": null,"nsiList": null,"fqdn": "ocnf-api-gateway.nef1-1:80","interPlmnFqdn": null,"ipv4Addresses": null,"ipv6Addresses": null ........
```

Refer to Get NEF Profile to get actual fqdn value.

Update the following:

4. Quit vim edit mode and save the changes. Wait for a while to verify whether NEF had been registered to NRF or not.

**Verifying kong API Gateway Service**

As the entrance of NEF all request would go through api gateway to specific service. Hence, it is necessary to verify if it works. If the following verification steps failed then contact Oracle support.

---

**Note:**
If no NRF deployed under current kubernetes cluster, then ignore the following settings under this section as there is no need for NEF to register itself to NRF service. The only way to invoke NEF services is to get API service URL manually then send request to it directly.
Refer to Enabling Loadbalancer with MetalLB for detailed description to update API gateway service to use Loadbalancer.

Run the following command and get API gateway service name `<NEF-Name>- api-gateway`.

```
kubectl get svc -n <NEF-Namespace>
```

The service must have an external IP allocated, and default administration port value 8001. Get the URL, `http://<API_GATEWAY_EXTERNAL_IP>:8001/services`.

Following is a sample content that displays:

```json
{
   "next":null,
   "data": [
   {
      "host": "<NAME>-pcf-nrf-clientservice",
      "created_at": 1553667178,
      "connect_timeout": 60000,
      "id": "897af1e0-8562-4843-b66a-e7453d29eeb9",
      "protocol": "http",
      "name": "nrf-client",
      "read_timeout": 60000,
      "port": 5910,
      "path": null,
      "updated_at": 1553667178,
      "retries": 5,
      "write_timeout": 60000
   },
   {
      "host": "istio-ingressgateway.istio-system.svc",
      "created_at": 1553667178,
      "connect_timeout": 60000,
      "id": "8c2b6ade-d876-4edd-ad28-c0fd28807a5c",
      "protocol": "http",
      "name": "sm-ingress",
      "read_timeout": 60000,
      "port": 80,
      "path": null,
      "updated_at": 1553667178,
      "retries": 5,
      "write_timeout": 60000
   },
   {
      "host": "<NAME>-pcf-bsf-management-service",
      "created_at": 1553667178,
      "connect_timeout": 60000,
      "id": "a3fa1ace-7016-4a2b-aabf-0e7e7a2618f0",
      "protocol": "http",
      "name": "bsf-service",
      "read_timeout": 60000,
      "port": 5903,
      "path": null,
      "updated_at": 1553667178,
      "retries": 5,
      "write_timeout": 60000
   },
   {
      "host": "<NAME>-pcf-pcf-amservice",
      "created_at": 1553667178,
      "connect_timeout": 60000,
      "id": "aa5ac6a2-162a-45ea-9a3c-c48f501b827e",
```
Use new URL http://<API_GATEWAY_EXTERNAL_IP>:8001/routes to check API gateway routes data.
"sources":null,
"destinations":null,
"snis":null,
"protocols": [  
  "http",
  "https"
],
"methods":null
},
{
  "created_at":1553667178,
  "updated_at":1553667178,
  "strip_path":true,
  "service":{
    "id":"aa5ac6a2-162a-45ea-9a3c-c48f501b827e"
  },
  "name":null,
  "hosts":null,
  "id":"52265a1f-9fcb-405f-ba10-1a1b3042a539",
  "preserve_host":false,
  "regex_priority":0,
  "paths":{
    "/*/am-service"
  },
  "sources":null,
  "destinations":null,
  "snis":null,
  "protocols": [  
  "http",
  "https"
],
  "methods":null
},
{
  "created_at":1553667178,
  "updated_at":1553667178,
  "strip_path":true,
  "service":{
    "id":"d020415d-6e50-4e35-87b5-050edblee6c"
  },
  "name":"sm-route",
  "hosts":null,
  "id":"5eec4c17-4b03-4bb1-a30d-1812121b8136",
  "preserve_host":false,
  "regex_priority":0,
  "paths":{
    "/*/sm-service"
  },
  "sources":null,
  "destinations":null,
  "snis":null,
  "protocols": [  
  "http",
  "https"
],
  "methods":null
},
{
  "created_at":1553667178,
  "updated_at":1553667178,
  "strip_path":true,
"service":{
  "id":"897af1e0-8562-4843-b66a-e7453d29eeb9",
  "name":null,
  "hosts":null,
  "id":"83c923a5-1c61-46af-b1b9-a7fa16ce3ce8",
  "preserve_host":false,
  "regex_priority":0,
  "paths":[
    "\n  "},
  "sources":null,
  "destinations":null,
  "snis":null,
  "protocols":[
    "http",
    "https"
  ],
  "methods":null
},
{ "created_at":1553667178,
  "updated_at":1553667178,
  "strip_path":true,
  "service":{
    "id":"8c2b6ade-d876-4edd-ad28-c0fd28807a5c",
    "name":null,
    "hosts":null,
    "id":"b42ee6c2-adc6-4ca5-b1ea-d90de1a78529",
    "preserve_host":false,
    "regex_priority":0,
    "paths":[
      "\n    "},
    "sources":null,
    "destinations":null,
    "snis":null,
    "protocols":[
      "http",
      "https"
    ],
    "methods":null
  },
{ "created_at":1553667178,
  "updated_at":1553667178,
  "strip_path":true,
  "service":{
    "id":"fe364cb8-2ff9-447f-88e6-f4e21b8d9022",
    "name":null,
    "hosts":null,
    "id":"b5134b99-b506-4e35-aa94-f4133daf3c33",
    "preserve_host":false,
    "regex_priority":0,
    "paths":[
      "\n    "},
    "sources":null,
    "destinations":null,
Known Behavior About Failed kong-migration Pod

For NEF deployment, kong-migration job startup quickly for first time deployment. However, it starts so quickly before the kong-database pod is ready, few kong-migration pod may fail until kong database pod is available. For example, see Figure 3-1.

![Figure 3-1 Kong-migration Pod](image)

Enabling Loadbalancer with MetalLB

Cloud Native Network have MetalLB installed, and free external IPs are already configured under MetalLB. Perform the following steps to enable LoadBalancer to specific services.

**Note:**

In NEF namespace, only API-Gateway service and cm service with GUI page requires load-balancer setting with accessible external IP. Other services are accessible by APIGateway service.

Updating API-Gateway Service

To update API-Gateway service:

1. Login to Kubernetes cluster master node using ssh command.
2. Run the following command to edit svc yaml file for API-Gateway:
   
   ```bash
   kubectl edit svc <NEF_NAME>-api-gateway -n <NEF_NAME_SPACE>
   ```

**Table 3-1 Variables**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEF_NAME</td>
<td>The --name value used in helm install command</td>
</tr>
</tbody>
</table>
### Table 3-1  (Cont.) Variables

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEF_NAME_SPACE</td>
<td>The <code>--namespace</code> value used in helm install command</td>
</tr>
</tbody>
</table>

Following is an sample content that displays in API-Gateway edit window

```yaml
1 # Please edit the object below. Lines beginning with a '#' will be ignored,
2 # and an empty file will abort the edit. If an error occurs while saving
this file will be
3 # reopened with the relevant failures.
4 #
5 apiVersion: v1
6 kind: Service
7 metadata:
  8 creationTimestamp: 2019-04-02T08:17:51Z
  9 labels:
    10 category: common
    11 io.kompose.service: <NEF_NAME>-pcf-api-gateway-service
  12 name: <NEF_NAME>-pcf-api-gateway-service
  13 namespace: <NEF_NAME_SPACE>
  14 resourceVersion: "25282719"
  15 selfLink: /api/v1/namespaces/<NEF_NAME_SPACE>/services/<NEF_NAME>-pcf-api-gateway-service
  16 uid: cec8f019-551f-11e9-acc3-a0369f714f30
  17 spec:
  18   clusterIP: 10.233.63.101
  19   externalTrafficPolicy: Cluster
  20   ports:
  21     - name: http
  22       nodePort: 32314
  23       port: 8080
  24       protocol: TCP
  25       targetPort: 8080
  26   selector:
  27     io.kompose.service: <NEF_NAME>-pcf-api-gateway-service
  28   sessionAffinity: None
  29   type: NodePort
  30 status:
  31   loadBalancer: {}  
```

3. Add two new lines after line 7, after metadata:
   annotations:
   ```yaml
   metallb.universe.tf/address-pool: <ADDRESS_POOL_NAME>
   ```

**Note:**

- As per user MetalLB setting, select an appropriate pool name to replace the variable, `<ADDRESS_POOL_NAME>`.
- annotation: line must be kept vertical align with line 16, while following line, `metallb.universe.tf/address-pool: <ADDRESS_POOL_NAME>` must be kept vertical align with line 10. If vertical align restriction failed to follow this rule, the svc yaml file update may fail.
4. Replace line 29 text, type: NodePort with type: LoadBalancer. Following is a sample content after replacing line 29.

```yaml
apiVersion: v1
category: common
group: <NEF_NAME>-pcf-api-gateway-service
kind: Service
metadata:
  creationTimestamp: 2019-04-02T08:17:51Z
  labels:
  - category: common
type: LoadBalancer

spec:
  clusterIP: 10.233.63.101
  externalTrafficPolicy: Cluster
  ports:
  - name: http
    nodePort: 32314
    port: 8080
    protocol: TCP
    targetPort: 8080
  selector:
    category: common
status:
  loadBalancer: {}
```

a. Quit vim editor and save changes. A new API-Gateway pod starts up. In the new pod, following sample content displays. Note that if the EXTERNAL-IP is available then the load balancer setting for API-Gateway service works.

```
NAME                     TYPE           CLUSTER-IP      EXTERNAL-IP
"<NEF_NAME>-pcf-api-gateway-service" LoadBalancer   10.xxx.xx.xx
TCP    4d
```

**Update cm-service**

Follow the same process logic to update svc yaml for `<NEF_NAME>-pcf-cm-service`. 
## Upgrading Network Exposure Function

User can perform the helm upgrade command in the following scenarios.

- Update an existing parameter setting, such as `global.imageTag`, `global.envMysqlHost`, etc.,
- Add more parameters per requirement

To update Binding Support Function services, execute the following command and specify the upgrade parameter:

```bash
helm upgrade <NAME>-nef \
--set global.envMysqlHost=<MYSQL_HOST>,global.envMysqlUser=nefusr,global.envMysqlPassword=nefpasswd \
--set global.envJaegerAgentHost=<JAEGER_SERVICE>.<JAEGER_SERVICE_NAMESPACE> \
--set platform.enabled=true,pcf.enabled=false,bsf.enabled=false,common.enabled=true,nef.enabled=true \
--set common.deploymentNrfClientService.envNamespace=<NAMESPACE>-nef,platform.appinfo.topic=NRF.UDM \
./<HELM_CHART_NAME_WITH_EXTENSION>
```

**Note:**

<dd>

- `<NAME>` and `<NAMESPACE>` must be same as helm install command.

**Note:**

The upgrade command is similar to install command, because, if user do not specify the same parameters for both upgrade and install, then the settings applied by install command may lost and use default settings from `values.yaml` file for missing parameters in upgrade command.

For specific deployments, the few parameters cannot be updated. Table 4-1 provides the details of parameters.
Table 4-1  Parameters for Deployment

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
</table>
| NEF        | platform.enabled=true,pcf.enabled=false,bsf.enabled=false,common.enabled=true,nef.enabled=true | For NEF deployment, following module must be enabled:  
  • platform  
  • nef  
  • common |
| NEF        | global.envManageNF=NEF,global.envSystemName=NEF,common.configmapApplicationConfig.nrf.ClientType=NEF | Since GUI service is an common service which defined under common module, it requires a startup parameter to show which NF should be showed under specific deployment. Same process logic applies to **nrfclient** service. |

**MetalLB Settings for Upgrade**

After executing the helm upgrade command, the configured MetalLB settings may lost. User is required to update the settings manually by following the procedure in the Enabling Loadbalancer with MetalLB.

**Verifying Upgrade**

Upgrade should ensure all pods under NEF namespace are updated based on the helm upgrade setting.

Execute below command:

```
kubectl get pod -n <NEF-Namespace>
```

Verify the following from the output of the above command:

- All pods under NEF namespace should either be in **Running** status or in **Completed** status. If any pod with error except for kong-migration pod status found, check pod log to view the reason for error.
- For updated service per helm upgrade setting, check its RESTART output and AGE output, if specific had been updated then the old pod should be killed and a new pod should start up. Hence, related RESTART value should be 0 and AGE value should be 3-4 seconds.
- API gateway service and CM service should support MetalLB setting.
Uninstalling Network Exposure Function

To uninstall or completely delete the Network Exposure Function deployment, execute the following command:

```
helm delete --purge <NEF-Name>
kubectl delete namespace <NEF-Namespace>
```

Verifying Uninstallation

To verify the NEF uninstallation, run the following command:

```
kubectl get namespace
```

**Result:** No NEF deployment should be found under the command outputs.