# Contents

1. **Introduction**
   - Overview 1-1
   - Acronyms 1-1
   - References 1-2
   - 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 Binding Support Function**
   - Pre-requisites 2-1
   - Installation Sequence 2-2
   - Creating Database Account on MySQL Database 2-2
   - Installation Preparation 2-4
   - Deploying Binding Support Function 2-5
   - Verifying Installation 2-9

3. **Configuring Binding Support Function**
   - Updating nrf Client Service Related configmap 3-1
     - Get nrf Service URL 3-2
     - Get BSF Profile 3-2
   - Verify kong API Gateway Service 3-2
     - Deploying API Gateway with Loadbalancer Type 3-3
     - Known Behavior About Failed kong-migration Pod 3-7
   - Enabling Loadbalancer with MetalLB 3-7
     - Updating API-Gateway Service 3-7
4 Upgrading Binding Support Function
   Verifying BSF Upgrade 4-2

5 Uninstalling Binding Support Function
   Verifying Uninstallation 5-1
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>Kong-migration Pod</td>
<td>3-7</td>
</tr>
</tbody>
</table>
List of Tables

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></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>Docket Images Name</td>
<td>2-6</td>
</tr>
<tr>
<td>2-2</td>
<td>Module Descriptions</td>
<td>2-6</td>
</tr>
<tr>
<td>2-3</td>
<td>Variable Names</td>
<td>2-7</td>
</tr>
<tr>
<td>2-4</td>
<td>Service Deployment Service Type</td>
<td>2-8</td>
</tr>
<tr>
<td>3-1</td>
<td>Variables</td>
<td>3-8</td>
</tr>
<tr>
<td>4-1</td>
<td>Parameters</td>
<td>4-2</td>
</tr>
</tbody>
</table>
1

Introduction

This document provides instructions to install Binding Support Function.

Overview

The Binding Support Function allows Policy Control Function to register, update, and remove the binding information from it, and allows Network Function (NF) consumers to discover the selected Policy Control Function.

The Binding Support Function stores the binding information for a certain PDU sessions and discovers the selected Policy Control Function according to the binding information. It also acts as diameter proxy agent or diameter redirect agent to Rx requests targeting an IP address of a UE to the selected Policy Control Function.

For any AF using Rx, such as P-CSCF, the Binding Support Function determines the selected Policy Control Function address according to the information carried by the incoming Rx requests.

Overview

Binding Support Function (BSF) provides a PDU session binding functionality, which ensures that an AF request for a certain PDU Session reaches the relevant PCF holding the PDU Session information. This service:

- Allows Policy Control Function users to register, update, and remove the binding information
- Allows NF consumers to retrieve the binding information

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>AF</td>
<td>Application Function</td>
</tr>
<tr>
<td>AMF</td>
<td>Access and Mobility Management Function</td>
</tr>
<tr>
<td>BSF</td>
<td>Binding Support Function</td>
</tr>
<tr>
<td>CNE</td>
<td>Cloud Native Environment</td>
</tr>
<tr>
<td>IWF</td>
<td>Inter-Working 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>NSSF</td>
<td>Network Slice Selection Function</td>
</tr>
</tbody>
</table>
### Table 1-1 (Cont.) Acronyms

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEI</td>
<td>Permanent Equipment Identifier</td>
</tr>
<tr>
<td>PCF</td>
<td>Policy Control Function</td>
</tr>
<tr>
<td>QFI</td>
<td>QoS Flow Identifier</td>
</tr>
<tr>
<td>SEPP</td>
<td>Security Edge Protection Proxy</td>
</tr>
<tr>
<td>SBI</td>
<td>Service Based Interface</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>

### References

Refer to the following documents for more information about cloud native binding support function.

- Cloud Native Environment Installation Document
- Binding Support Function Cloud Native User’s Guide

### 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 Binding Support Function

This section provides instructions for installing Binding Support Function.

Pre-requisites

Following are the pre-requisites before installing Binding Support Function.

BSF Software

BSF software includes:

<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-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. Install the software before proceeding with BSF installation.

Note:

If you are using NRF, install it before proceeding with the BSF 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 Binding Support Function helm charts are available
- Local docker image repository where the Oracle Communications Binding Support Function images are available

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 recommend 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 BSF installation.

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

Creating Database Account on MySQL Database

Create a new database account on MySQL database by executing following command:
CREATE USER 'bsfusr'@'%' IDENTIFIED BY 'bsfpasswd';
GRANT ALL PRIVILEGES ON *.* TO 'bsfusr'@'%;

Login to MySQL console as a new user created above:

```
mysql -h<MYSQL_HOST> -u<USERNAME> -p<PASSWORD>
```

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

```
mysql -h<MYSQL_HOST> -ubsfusr -pbsfpasswd
```

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_gd6b0a6ndpxc55qbibre2cldc' (`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);
Installation Preparation

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

1. Download the BSF package file from Oracle Software Delivery Cloud (OSDC). Package is named as follows:

   `<nfname>-pkg-<marketing-release-number>.tgz`

   For example, `ocbsf-pkg-1.0.0.0.0.tgz`

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

- **BSF Docker Images File:**
  ocbsf-images-1.0.0.tar

- **Helm File:**
  ocbsf-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 Binding Support Function

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

To deploy BSF:

1. Download the file, **ocbsf-pkg-1.0.0.0.0.tgz**.
2. Untar **ocbsf-pkg-1.0.0.0.0.tgz**.
3. Untar displays the following files:
   
<table>
<thead>
<tr>
<th>ocbsf-pkg-1.0.0.0.0.tgz</th>
</tr>
</thead>
<tbody>
<tr>
<td>ocbsf-1.0.0.tgz (helm chart to be used in step 4)</td>
</tr>
<tr>
<td>ocbsf-images-1.0.0.tar (docker images)</td>
</tr>
<tr>
<td>Readme.txt (Contains cksum and md5sum of tarballs)</td>
</tr>
</tbody>
</table>

4. Check the checksums of tarballs mentioned in the **Readme.txt** file.
5. Run the following command to load **ocbsf-images-1.0.0.tar** to docker and push imported docker images to user docker registry.

```
docker load --input /<IMAGE_PATH>/ocbsf-images-1.0.0.tar

docker tag ocpcf/bsf_management_service:1.0.0 <customer repo>/bsf_management_service:1.0.0

docker push <customer repo>/bsf_management_service: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 details of docker images file names.

**Table 2-1   Docket Images Name**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Service Name</th>
<th>Docket Image Name</th>
<th>Service Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DB Service</td>
<td>db-service</td>
<td>Common</td>
</tr>
<tr>
<td>2</td>
<td>Nrf Client Service</td>
<td>nrf_clientservice</td>
<td>Common</td>
</tr>
<tr>
<td>3</td>
<td>CM Service</td>
<td>ocpm_cm_service</td>
<td>Common</td>
</tr>
<tr>
<td>4</td>
<td>Performance Monitoring Service</td>
<td>perf_info</td>
<td>Platform</td>
</tr>
<tr>
<td>5</td>
<td>Config Server Service</td>
<td>ocpm_config_server</td>
<td>Common</td>
</tr>
<tr>
<td>6</td>
<td>BSF Service</td>
<td>bsf_management_service</td>
<td>BSF</td>
</tr>
<tr>
<td>7</td>
<td>Diameter Gateway</td>
<td>diam-gateway</td>
<td>BSF</td>
</tr>
<tr>
<td>8</td>
<td>Diameter Connector</td>
<td>diam-connector</td>
<td>BSF</td>
</tr>
<tr>
<td>9</td>
<td>Application Info Service</td>
<td>app_info</td>
<td>Platform</td>
</tr>
<tr>
<td>10</td>
<td>Readiness Check</td>
<td>readiness-detector</td>
<td>Common</td>
</tr>
<tr>
<td>11</td>
<td>Oracle Linux 7 with JDK11</td>
<td>ocpm_ol7_jdk11</td>
<td></td>
</tr>
</tbody>
</table>

Table 2-2 provides 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, API Gateway</td>
</tr>
<tr>
<td>platform</td>
<td>Platform service to provide monitoring services, also shared by separate NF</td>
</tr>
<tr>
<td>bsf</td>
<td>Define Binding Support Function services</td>
</tr>
</tbody>
</table>

6. Navigate to helm chart and execute the following command:
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.

```
helm install --namespace=<NAMESPACE>-bsf-1 --name=<NAME>-bsf-1 \
--set
    global.envMysqlHost=<MYSQL_HOST>, global.envMysqlUser=bsfusr, global.envMysqlPassword=bsfpasswd \
--set global.envJaegerAgentHost=<JAEGER_SERVICE>.<JAEGER_SERVICE_NAMESPACE> \
--set
    global.envManageNF=BSF, global.envSystemName=BSF-1, common.configmapApplicationConfig.nrfClientType=BSF-1 \
--set
    global.imageTag=<IMAGE_TAG>, global.dockerRegistry=<DOCKER_REGISTRY_ADDRESS> \
--set
    platform.enabled=true, pcf.enabled=false, bsf.enabled=true, common.enabled=true, ncf.enabled=false \
--set common.deploymentNrfClientservice.envNamespace=$<NAMESPACE>-bsf-1 \
./<HELM_CHART_NAME_WITH_EXTENSION>
```

Table 2-3 provides the details of the variables

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NAMESPACE&gt;</td>
<td>Deployment NF namespace used by helm command</td>
<td>Allowed maximum character length is 10. Variable can contain upper case, and lower case alphabets, numbers, and special characters, _, -.</td>
</tr>
<tr>
<td>&lt;NAME&gt;</td>
<td>Deployment NF name used by helm command</td>
<td>-</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>
</tbody>
</table>
Table 2-3  (Cont.) Variable Names

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;JAEGGER_SERVICE&gt;</td>
<td>Both parameters could be found in same Kubernetes cluster</td>
</tr>
<tr>
<td>&lt;JAEGGER_SERVICE_NAMESPACE&gt;</td>
<td>Follow the below format:</td>
</tr>
<tr>
<td></td>
<td>&lt;JAEGGER_AGENT_SERVICE_NAME&gt;.&lt;JAEGGER_NAMESPACE&gt;</td>
</tr>
<tr>
<td></td>
<td>For example, consider OCCNE. Run the command, kubectl get svc -n occne-infra to get all services under namespace occne-infra.</td>
</tr>
<tr>
<td></td>
<td>Consider *-jaeger-agent as jaeger service name, such as occne-tracer-jaeger-agent.occne-infra, then the jaeger agent service name under jaeger deployment is occne-tracer-jaeger-agent.</td>
</tr>
<tr>
<td></td>
<td>Example: occne-tracer-jaeger-agent.occne-infra</td>
</tr>
</tbody>
</table>

| <IMAGE_TAG> | The image tag used in customer docker registry. It is recommended to use same image tag when pull docker image to registry. If followed above steps to push docker image to customer docker registry, then the <IMAGE_TAG> value should be 1.0.0 |
|            | 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 |

| <DOCKER_REGISTRY_ADDRESS> | Customer docker registry address |
|                         | If registry has port value, add port. For example, reg-1:5000 |

Table 2-4 provides Service Deployment Service Type.

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. 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, &lt;NodeIP&gt;:&lt;NodePort&gt; Most of the PCF services use NodePort to deploy.</td>
</tr>
</tbody>
</table>
Table 2-4  (Cont.) Service Deployment Service Type

| LoadBalancer | 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. |

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 <BSF-Namespace>
kubectl get pod -n <BSF-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.
Configuring Binding Support Function

User can perform the following configurations:

**Updating nrf Client Service Related configmap**

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

- Currently deployed BSF uses updated configmap to register BSF itself to NRF service.
- Client gets BSF related FQDN, IPv4, or IPv6 address from NRF service and send request to specific service under BSF with that FQDN, IPv4 or IPv6 address. The request reaches to API Gateway service under BSF deployment.
- The API Gateway service under BSF 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 BSF, in order to register BSF to NRF services and hence it is mandatory to update configmap to use correct nrf service URL and apply some extra settings.

---

**Note:**

If no NRF deployed under current kubernetes cluster, then ignore the following settings under this section as there is no need for BSF to register itself to NRF service. The only way to invoke BSF services is to get API service URL manually then send request to it directly.

---

Before updating nrf client service, perform the following:

- Get nrf Service URL
- Get BSF Profile

To update nrf client service:

1. Run the following command and get config map item, `<BSF-NAME>-application-config`:
   ```bash
   kubectl get configmap -n <BSF-Namespace>
   ```
2. Run the following command and then navigate to vim editor with config map content displayed
   ```bash
   kubectl edit configmap <BSF-NAME>-application-config -n <BSF-Namespace>
   ```
3. Update `nrfApiRoot` to actual deployed nrf service URL. For example, `nrfApiRoot=http://ocnrf-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 ipv6adress should be populated in the NF Profile under `configmap` definition. NF Profile here indicates the
entrance of current deployed BSF 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 `ocbsf-api-gateway.bsf1-1:80` for fqdn value. Following result displays:

```json
appProfiles=[{"nfInstanceId": "fe7d882b-0541-4c7d-ab84-c6d70b1b0123","nfType": "PCF","nfStatus": "REGISTERED","plmn": null,"nsiList": null,"fqdn": "ocbsf-api-gateway.bsf1-1:80","interPlmnFqdn": null,"ipv4Addresses": null,"ipv6Addresses": null,......
```

Refer to Get BSF Profile to get actual fqdn value.

4. Set

```
"ipEndPoints": [{"ipv4Address": "ocpm-staging-bsf-2-api-gateway.ocpm-staging-bsf-2.svc","ipv6Address": null,"transport": "TCP","port": 5903}]
```

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

### 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 BSF Profile

The BSF Profile here indicates the entrance of current deployed BSF 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, `<BSF-Name> pcf-api-gateway` with default application port value 80.

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

For example, consider `<BSF-Name>` variable, as `bsf1-1` and `API gateway service` name as `ocbsf-api-gateway`, then the `API gateway service` value is `ocpcf-api-gateway.pcf1:80`.

### Verify kong API Gateway Service

As the entrance of BSF, all request goes through API gateway to specific service. It is necessary to verify if it works.
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 `<BSF-Name>- pcf-api-gateway`.

```bash
kubectl get svc -n <BSF-Namespace>
```

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

**Deploying API Gateway with Loadbalancer Type**

To deploy API Gateway with Loadbalancer type:

- Get external IP of API Gateway service and open the link, http://<API_GATEWAY_EXTERNAL_IP>:8001/services via browser. Following is a sample content that displays:

```json
{
    "next":null,
    "data":{
        "host":"
```
• Use new URL http://<API_GATEWAY_EXTERNAL_IP>:8001/routes to check API gateway routes data.
Chapter 3
Verify kong API Gateway Service

```json
{
    "created_at":1553667178,
    "updated_at":1553667178,
    "strip_path":true,
    "service":{
        "id":"897af1e0-8562-4843-b66a-e7453d29ebeb9"
    },
    "name":null,
    "hosts":null,
    "id":"83c923a5-1c61-46af-b1b9-a7fa16ce3ce8",
    "preserve_host":false,
    "regex_priority":0,
    "paths": [
        "/nrf-client"
    ],
    "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-blea-d90dela78529",
    "preserve_host":false,
    "regex_priority":0,
    "paths": [
        "/sm-ingress"
    ],
    "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":"f364cb8-2ff9-447f-88e6-f4e21b8d9022"
}
```
Known Behavior About Failed kong-migration Pod

For BSF 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

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 BSF 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 <BSF_NAME>-pcf-api-gateway-service -n <BSF_NAME_SPACE>
   ```

### Table 3-1 Variables

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

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

```yaml
# Please edit the object below. Lines beginning with a '#=' will be ignored, and an empty file will abort the edit. If an error occurs while saving this file will be reopened with the relevant failures.

apiVersion: v1
kind: Service
metadata:
  creationTimestamp: 2019-04-02T08:17:51Z
  labels:
    category: common
    io.kompose.service: <BSF_NAME>-pcf-api-gateway-service
  name: <BSF_NAME>-pcf-api-gateway-service
  namespace: <BSF_NAME_SPACE>
  resourceVersion: "25282719"
  selfLink: /api/v1/namespaces/<BSF_NAME_SPACE>/services/<BSF_NAME>-pcf-api-gateway-service
  uid: cec8f019-551f-11e9-acc3-a0369f714f30
spec:
  clusterIP: 10.233.63.101
  externalTrafficPolicy: Cluster
  ports:
    - name: http
      nodePort: 32314
      port: 8080
      protocol: TCP
      targetPort: 8080
  selector:
    io.kompose.service: <BSF_NAME>-pcf-api-gateway-service
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}
```

3. Add two new lines after line 7, after metadata:

   ```yaml
   annotations:
     metallb.universe.tf/address-pool: <ADDRESS_POOL_NAME>
   ```
4. Replace line 29 text, **type: NodePort** with **type: LoadBalancer**.

   Following is a sample content after replacing line 29.

```yaml
# Please edit the object below. Lines beginning with a '#' will be ignored, 
# and an empty file will abort the edit. If an error occurs while saving 
# this file will be 
# reopened with the relevant failures.

apiVersion: v1
type: LoadBalancer

kind: Service
metadata:
  creationTimestamp: 2019-04-02T08:17:51Z
  labels:
    category: common
    io.kompose.service: <BSF_NAME>-pcf-api-gateway-service
    name: <BSF_NAME>-pcf-api-gateway-service
  namespace: <BSF_NAME_SPACE>
  resourceVersion: "25282719"
  selfLink: /api/v1/namespaces/<BSF_NAME_SPACE>/services<BSF_NAME>-pcf-api-gateway-service
  uid: cec8f019-551f-11e9-acc3-a0369f714f30
  annotations:
    metallb.universe.tf/address-pool: <ADDRESS_POOL_NAME>
spec:
  clusterIP: 10.233.63.101
  externalTrafficPolicy: Cluster
  ports:
    - name: http
code: NodePort: 32314
  port: 8080
  protocol: TCP
  targetPort: 8080
  selector:
    io.kompose.service: <BSF_NAME>-pcf-api-gateway-service
  sessionAffinity: None
  type: LoadBalancer
status:
  loadBalancer: {}
```

5. 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.

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
</tr>
</thead>
</table>
Updating cm-service

Follow the same process logic to update svc yaml for `<BSF_NAME>-pcf-cm-service`.
Upgrading Binding Support Function

You can apply the helm upgrade command in the following scenarios:

- Update existing parameter setting. For example, `global.imageTag`, `global.envMysqlHost`
- Add new parameters as per the requirement

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

Following is a sample execution command is given below:

```bash
helm upgrade <NAME>-bsf-1 \
   --set global.envMysqlHost=<MYSQL_HOST>,global.envMysqlUser=bsfusr,global.envMysqlPassword=bsfpasswd \
   --set global.envJaegerAgentHost=<JAEGGER_SERVICE>.<JAEGGER_SERVICE_NAMESPACE> \
   --set global.envManageNF=BSF,global.envSystemName=BSF-1,common.configmapApplicationConfig.nrfClientId=BSF-1 \
   --set global.imageTag=<IMAGE_TAG>,global.dockerRegistry=<DOCKER_REGISTRY_ADDRESS> \
   --set platform.enabled=true,pcf.enabled=false,bsf.enabled=true,common.enabled=true,nef.enabled=false \
   --set common.deploymentNrfClientservice.envNamespace=${NAMESPACE}-bsf-1 \
   ./<HELM_CHART_NAME_WITH_EXTENSION>
```

**Note:**

*<NAME>* and *<NAMESPACE>* must be same as in 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 deployment, few parameters cannot be updated. *Table 4-1* provides details of the parameters which cannot be updated.
Table 4-1 Parameters

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| BSF        | platform.enabled=true,pcf.enabled=false, bsf.enabled=true,common.enabled=true, nef.enabled=false | For deployment, below module must be enabled:  
• platform  
• bsf  
• common |
| BSF        | global.envManageNF=BSF,global.envSystemName=BSF-1,common.configmapApplicationConfig.nrfClientType=BSF-1 | Since GUI service is an common service which defined under common module, it requires the startup parameters to show which NF should be displayed 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 BSF Upgrade

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

Execute below command:

```bash
kubectl get pod -n <BSF-Namespace>
```

Verify the following from the output of the above command:

- All pods under BSF 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 Binding Support Function

To uninstall or completely delete the Binding Support Function deployment, execute one of the following command:

```
helm del --purge <helm_release_name_for_ocbsf> or
kubectl delete namespace <BSF-Namespace>
```

Verifying Uninstallation

To verify the BSF uninstallation, run the following command:

```
kubectl get namespace
```

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