Oracle® Communications

Cloud Native Binding Support Function Installation Guide

Release 1.4

F26335-03

May 2020

Oracle Communications Cloud Native Binding Support Function Installation Guide, Release 1.4

F26335-03

Contents

What's New in This Guide

This section introduces the new/updated features in Oracle Communications Cloud Native Binding Support Function Installation Guide.

New/Updated Features

There are no documentation updates in this guide. This guide has been republished as part of the bundled documentation offering for Cloud Native Core Release 2.2.0.

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:

- 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**.
- For Hardware, Networking and Solaris Operating System Support, select **3**.

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.

1 Introduction

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

The BSF 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.

The BSF provides a PDU session binding functionality, which ensures that an Application Function (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

For more information, see Oracle Communications Cloud Native Binding Support Function User's Guide.

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

Acronyms and Terminology

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

Table 1-1 Acronyms and Terminology

Acronym	Definition
5GC	5G Core Network
5GS	5G System
5G-AN	5G Access Network
5G-EIR	5G-Equipment Identity Register
5G-GUTI	5G Globally Unique Temporary Identifier

Table 1-1 (Cont.) Acronyms and Terminology

Acronym	Definition
5G-S-TMSI	5G S-Temporary Mobile Subscription Identifier
5QI	5G QoS Identifier
AF	Application Function
AMF	Access and Mobility Management Function
AS	Access Stratum
AUSF	Authentication Server Function
BSF	Binding Support Function
CAPIF	Common API Framework for 3GPP northbound APIs
CUSTOMER_REPO	The docker registry address in customer side, plus Port No. if registry has port attached
IMAGE_TAG	The image tag from release tar file is 1.4.0, You can decide to use any tag No. Then push related docker image with that specific tag to their registry
METALLB_ADDRESS_POOL	The address pool which configured on metallb to provide external IPs
MPS	Multimedia Priority Service
N3IWF	Non-3GPP InterWorking Function
NAI	Network Access Identifier
NEF	Network Exposure Function
NF	Network Function
NGAP	Next Generation Application Protocol
NR	New Radio
NRF	Network Repository Function
NSI ID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
NSSP	Network Slice Selection Policy
NWDAF	Network Data Analytics Function
PCF	Policy Control Function
PDR	Packet Detection Rule
PEI	Permanent Equipment Identifier
PER	Packet Error Rate
PFD	Packet Flow Description
PPD	Paging Policy Differentiation
PPF	Paging Proceed Flag
PPI	Paging Policy Indicator
PSA	PDU Session Anchor
QFI	QoS Flow Identifier
QoE	Quality of Experience
(R)AN	(Radio) Access Network
RQA	Reflective QoS Attribute
RQI	Reflective QoS Indication

Table 1-1 (Cont.) Acronyms and Terminology

Acronym	Definition
SA NR	Standalone New Radio
SBA	Service Based Architecture
SBI	Service Based Interface
SD	Slice Differentiator
SEAF	Security Anchor Functionality
SEPP	Security Edge Protection Proxy
SMF	Session Management Function
SMSF	Short Message Service Function
S-NSSAI	Single Network Slice Selection Assistance Information
SSC	Session and Service Continuity
SSCMSP	Session and Service Continuity Mode Selection Policy
UDM	Unified Data Management
UDR	Unified Data Repository
UDSF	Unstructured Data Storage Function

2
Planning Your Installation

This chapter describes how to plan your installation of the Binding Support Function (BSFF) cloud native deployment package.

Overview

Binding Support Function (BSF) cloud native deployment package includes ready-to-use images and Helm charts to help you orchestrate containers in Kubernetes.

You can use the Docker images and Helm chart to help you deploy and manage Pods of BSF product services in Kubernetes. Communication between Pods of services of BSF products are preconfigured in the Helm charts.

Below table lists the docker images for BSF.

Table 2-1 Docker Images for BSF

S.No	Service Name	Docker Image Name
1	Application info service	app_info
2	BSF Service	bsf_management_service
3	CM Service	ocpm_cm_service
4	Config Server Service	ocpm_config_server
5	Diameter Connector	diam-connector
6	Diameter Gateway	diam-gateway
7	Performance Monitoring Service	perf_info
8	Readiness check	readiness-detector

Software Compatibility

The following softwares must be installed before installing Binding Support Function (BSF):

Software	Version
Kubernetes	v1.15.3
HELM	v2.14.3

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

Software	App Version	Notes
alertmanager	0.18.0	Needed for Tracing Area
elasticsearch	6.8.2	Needed for Logging Area
elastic-curator	5.5.4	Needed for Logging Area
elastic-exporter	1.0.2	Needed for Logging Area
logs	2.0.7	Needed for Logging Area
kibana	6.7.0	Needed for Logging Area
grafana	6.1.6	Needed for Metrics Area
prometheus	9.1.2	Needed for Metrics Area
prometheus-node-exporter	0.17.0	Needed for Metrics Area
metallb	0.7.3	Needed for External IP
metrics-server	0.3.1	Needed for Metric Server
occne-snmp-notifier	0.3.0	Needed for Metric Server
tracer	0.8.3	Needed for Tracing Area

Note:

The above softwares are available if the BSF is deployed in the Oracle Communications Cloud Native Environment (OCCNE). If you are deploying BSF in any other environment, the above softwares must be installed before installing the BSF. To check the installed software items,

```
helm ls
```

Some of the systems may need to use helm command with **admin.conf** file as follows:

```
helm --kubeconfig admin.conf
```

Environment Setup

Note:

This section is applicable only when the Binding Support Function (BSF) is deployed in the environment, other than OCCNE.

Network access

The Kubernetes cluster hosts must have network access to:

- Local helm repository, where the BSF helm charts are available.

To check if the Kubernetes cluster hosts have network access to the local helm repository, execute the following command:

```
helm repo update
```

> **Note:**
>
> Some of the systems may need to use helm command with **admin.conf** file as follows:
>
> ```
> helm --kubeconfig admin.conf
> ```

- Local docker image repository, where the BSF images are available.
  To check if the Kubernetes cluster hosts have network access to the local docker image repository, pull any image with tag name to check connectivity by executing the following command:

  ```
  docker pull docker-repo/image-name:image-tag
  ```

  where:

  docker-repo is the IP address or host name of the repository.

  image-name is the docker image name.

  image-tag is the tag the image used for the BSF pod.

> **Note:**
>
> All the kubectl and helm related commands that are used in this guide must be executed on a system depending on the infrastructure/deployment. It could be a client machine, such as, a VM, server, local desktop, and so on.

Client Machine Requirements

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

- It should have network access to the helm repository and docker image repository.
- 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 with the **push** plugin. The environment should be configured so that the `helm install` command deploys the software in the Kubernetes cluster.

Server or Space Requirements

For information on the server or space requirements, see the *Oracle Communications Cloud Native Environment (OCCNE) Installation Guide*.

Configuring Database

Introduction

Binding Support Function (BSF) microservices use MySQL database to store the configuration and run time data. Following microservices require dedicated MySQL databases created in MySQL data tier.

- BSF Management Service - To store Binding session state
- Config Server - To store configuration data

The BSF requires the database administrator to create user in MySQL DB and provide necessary permissions to access the databases. Before installing the BSF it is required that the MySQL user and databases are created.

Customizing Database Name

Each microservice has a default database name assigned as mentioned in below table:

Service Name	Default Database Name
BSF Management Service	ocpm_bsf
Config Server Service	ocpm_config_server

It is recommended to use unique database name when there are multiple instances of BSF deployed in the network and they share the same data tier (MySQL cluster).

It is recommended to create custom unique database name, by simply prefixing the default database name with an identifier that uniquely identifies this instance of BSF deployment. This way database name uniqueness can be achieved across all deployments. However, you can use any prefix to create the unique database name. For example, if the OCBSF deployment name is "site1" then the BSF Management Service database can be named as " site1_ocpm_bsf". Refer the BSF Customization Parameters section for how to override default database names with custom database names.

Configuring MYSQL Database

The BSF deployment requires the database administrator to create a user in the MYSQL database, and to assign it to the group having necessary permissions to access the tables on all SQL nodes.

> **Note:**
>
> This procedure must be performed only once before initiating the BSF deployment.

To configure MYSQL database for the different microservices:

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 tables for the different microservices:

> **Note:**
>
> User with admin privileges can create the tables.

BSF Management Service

In the following script default database name " ocpm_bsf" is used, if custom database name is used then replace it wherever applicable.

```
CREATE DATABASE IF NOT EXISTS `ocpm_bsf`;

CREATE TABLE IF NOT EXISTS ocpm_bsf.pcf_binding (
  binding_id binary(16) not null,
  ipv4_addr varchar(64),
  ip_domain varchar(128),
  ipv6_prefix varchar(64),
  mac_addr_48 varchar(64),
  dnn varchar(128),
  supi varchar(64),
  gpsi varchar(64),
  snssai_sd varchar(64),
  snssai_sst integer,
  created_date_time datetime(6) not null,
  json_content longblob not null,
  primary key (binding_id),
  key idx_created_date_time (created_date_time),
  key idx_ipv4Addr (ipv4_addr, created_date_time),
  key idx_ipv6Prefix (ipv6_prefix, created_date_time),
  key idx_macAddr48 (mac_addr_48, created_date_time),
  key idx_supi (supi, created_date_time),
  key idx_gpsi (gpsi, created_date_time)
);
```

Config Server Service

In the following script default database name " ocpm_config_server" is used, if custom database name is used then replace it wherever applicable.

```
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;
```

5. Create the user and assign it to a group having necessary permissions to access the tables on all of the SQL nodes by executing the following command: :

   ```
   CREATE USER 'username'@'%' IDENTIFIED BY 'password';
   ```

   where:
   username is the username and *password* is the password for MySQL database user.

   For Example: In the below example, "bsfusr" is used as username, "bsfpasswd" is used as password.

   ```
   CREATE USER 'bsfusr'@'%' IDENTIFIED BY 'bsfpasswd';
   GRANT SELECT, INSERT, UPDATE, DELETE ON ocpm_bsf.* TO 'bsfusr'@'%';
   GRANT SELECT, INSERT, UPDATE, DELETE ON ocpm_config_server.* TO
   'bsfusr'@'%';
   ```

6. Execute the command, `show databases`, to view database.
7. Exit from database and logout from MYSQL node.

3

Installing the Binding Support Function Cloud Native Deployment Package

This chapter describes how to install the BSF cloud native deployment package on a cloud native environment.

Installing Binding Support Function Cloud Native Services

To install Binding Support Function (BSF) cloud native services, do this:

1. Go to the Oracle software delivery Web site:
   http://edelivery.oracle.com
2. Sign in with your user name and password.
3. Search for Binding Support Function and select the BSF software pack from the search results.
4. Download the BSF package file. Package is named as follows:
   ReleaseName-pkg-*Releasenumber*.tgz

   where:

   ReleaseName is a name which is used to track this installation instance.

   Releasenumber is the release number.

   For example, `ocbsf-pkg-1.4.0.0.0.tgz`
5. Untar the BSF package file to get BSF docker image tar file.
   `tar -xvzf` *ReleaseName*-pkg-*Releasenumber*.tgz

   This command results into *ReleaseName*-pkg-*Releasenumber* directory.

   The directory consists of the following:

   - **BSF Docker Images File:**
     `ocbsf-images-1.4.0.tar`
   - **Helm File:**
     `ocbsf-1.4.0.tgz`
   - **Readme txt File:**
     `Readme.txt` (Contains cksum and md5sum of tarballs)
6. Load the **ocbsf-images-1.4.0.tar** file into the Docker system

   ```
   docker load --input /IMAGE_PATH/ocbsf-images-1.4.0.tar
   ```
7. Verify that the image is loaded correctly by entering this command:

   ```
   docker images
   ```

8. Create a new tag for each imported image and push the image to the customer docker registry by entering this command:

```
docker tag ocbsf/bsf_management_service:1.4.0 CUSTOMER_REPO/
bsf_management_service:1.4.0
docker push CUSTOMER_REPO/bsf_management_service:1.4.0

docker tag ocbsf/diam-connector:1.4.0 CUSTOMER_REPO/diam-connector:1.4.0
docker push CUSTOMER_REPO/diam-connector:1.4.0

docker tag ocbsf/diam-gateway:1.4.0 CUSTOMER_REPO/diam-gateway:1.4.0
docker push CUSTOMER_REPO/diam-gateway:1.4.0

docker tag ocbsf/readiness-detector:1.4.0 CUSTOMER_REPO/readiness-
detector:1.4.0
docker push CUSTOMER_REPO/readiness-detector:1.4.0

docker tag ocbsf/ocpm_cm_service:1.4.0 CUSTOMER_REPO/ocpm_cm_service:
1.4.0
docker push CUSTOMER_REPO/ocpm_cm_service:1.4.0

docker tag ocbsf/nrf_clientservice:1.4.0 CUSTOMER_REPO/
nrf_clientservice:1.4.0
docker push CUSTOMER_REPO/nrf_clientservice:1.4.0

docker tag ocbsf/ocpm_config_server:1.4.0 CUSTOMER_REPO/
ocpm_config_server:1.4.0
docker push CUSTOMER_REPO/ocpm_config_server:1.4.0

docker tag ocbsf/perf_info:1.4.0 CUSTOMER_REPO/perf_info:1.4.0
docker push CUSTOMER_REPO/perf_info:1.4.0

docker tag ocbsf/app_info:1.4.0 CUSTOMER_REPO/app_info:1.4.0
docker push CUSTOMER_REPO/app_info:1.4.0


docker tag ocbsf/apigateway:1.0.0 CUSTOMER_REPO/ocpm_apigateway:1.4.0
docker push CUSTOMER_REPO/ocpm_apigateway:1.4.0
```

where:

CUSTOMER_REPO is the docker registry address having Port Number, if registry has port attached.

> **Note:**
>
> For OCCNE, copy the package to bastion server and use **localhost: 5000** as CUSTOMER_REPO to tag the images and push to bastion docker registry.

> **Note:**
>
> You may need to configure the Docker certificate before the push command to access customer registry via HTTPS, otherwise, docker push command may fail.

9. Create the customize *ocbsf-custom-values-1.4.0.yaml* file with the required input parameters. To customize the file, see Customizing Binding Support Function .
10. Go to the helm chart directory:

    ```
    cd ocbsf-pkg-1.4.0.0.0
    ```

11. Install BSF by entering this command:

    ```
    helm install HELM_CHART_PATH/bsf-1.4.0.tgz --name BSF_NAME --namespace
    BSF_NAMESPACE -f CUSTOM_VALUES_YAML_FILE
    ```

    where:
    *HELM_CHART_PATH*is the location of the helm chart extracted from ocbsf-pkg-1.4.0.tgz file

    BSF_NAME is the release name used by helm command.

    BSF_NAMESPACE is the deployment namespace used by helm command.

    CUSTOM_VALUES_YAML_FILE - is the name of the custom values yaml file (including location).

    For example:

    ```
    helm install /home/cloud-user/bsf-1.4.0.tgz --name ocbsf --namespace
    ocbsf -f ocbsf-custom-values-1.4.0.yaml
    ```

12. Check the deployment status by entering this command

    ```
    helm status --name BSF_NAME
    ```

    where:

    BSFNAME is the release name used by helm command.

    For example:

    ```
    helm status --name ocbsf
    ```

    You will see the status as **DEPLOYED** if the deployment has been done successfully.
    Execute the following command to get status of pods

    ```
    kubectl get pod -n BSF_NAMESPACE
    ```

    where:

BSF_NAMESPACE is the name space in which to create BSF Kubernetes objects. All the BSF microservices are deployed in this kubernetes namespace.

For example:

```
kubectl get pod -n ocbsf
```

You will see the status as **Running** for all the nodes if the deployment has been done successfully.

4
Customizing Binding Support Function

This chapter describes how to customize the Binding Support Function (BSF) deployment in a cloud native environment.

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

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

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

Following is a sample **ocbsf-custom-values-1.4.0.yaml** file created based on all the parameters described in the Binding Support Function Customization Parameters section .

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


# section:- global
global:
   # Docker registry name
   dockerRegistry: 'occne-bastion:5000'
   #image tag for readiness detector
   imageServiceDetector: readiness-detector:1.4.0
   #Jaeger Hostname
   envJaegerAgentHost: 'occne-tracer-jaeger-agent.occne-infra'

   # Primary MYSQL Host IP or Hostname
   envMysqlHost: ''
   # OCBSF MYSQL UserName
   envMysqlUser: ''
   # OCBSF MYSQL Password
   envMysqlPassword: ''

   # To assign custom node port for CM service(use port range above 30000)
   cmServiceNodePort: 0
```

```
   # To assign custom node port for Diameter Gateway service(use port
range above 30000)
   bsfDiamGatewayNodePort: 0

   #Network Function Name
   envManageNF: 'BSF'
   #System Name
   envSystemName: 'BSF'

bsf:
  enabled: true
  deploymentBsfManagementService:
    # MySQL Database for Bsf Management Service
    envMysqlDatabase: 'ocpm_bsf'
common:
  deploymentPcfCmService:
    # MySQL Database for session viewer. Fill the value of Bsf Management
Service database name here
    envMysqlDatabaseSessionViewer: 'ocpm_bsf'
  deploymentPcfConfig:
    # MySQL Database for Config Server
    envMysqlDatabase: 'ocpm_config_server_bsf'
  deploymentNrfClientservice:
    envNfNamespace: 'bsf-rc-4'
    envNfType: 'bsf'
    envConsumeSvcName: 'bsf-management-service:nbsf-management'
  configmapApplicationConfig:
    # IpAddress/Fqdn of NRF
    nrfHost: '10.75.153.80'
    nrfServicePort: 31871
    nrfClientType: BSF
api-gateway:
  fullnameOverride : 'bsf-apigateway'
  image:
    registry: 'occne-bastion:5000'
  service:
    type: NodePort
    configServiceNetworkNameEnabled: false
    configServiceNetworkName: 'metallb.universe.tf/address-pool: oam'
```

5
Binding Support Function Customization Parameters

> **Note:**
>
> - All parameters mentioned as mandatory must be present in custom values file.
> - All fixed value parameters mentioned must be present in the custom values file with the exact values as specified here.

Global Configurations

These configuration parameters are common for all micro services.

Table 5-1 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Notes
dockerRegistry	Name of the Docker registry which hosts Binding Support Function (BSF) docker images	Yes	Not applicable	This is a docker registry running in OCCNE bastion server where all BSF docker images will be loaded. For example, 'occne-bastion: 5000'
envMysqlHost	IP address or host name of the MySql server which hosts BSF's databases	Yes	Not applicable	
envMysqlUser	Mysql Username to access BSF's databases	Yes	Not applicable	
envMysqlPassword	Mysql password for the username	Yes	Not applicable	

Table 5-1 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Notes
envJaegerAgentHost	Hostname or IP address for the Jaeger Agent	Yes	Not applicable	This is the fqdn of Jaeger Agent service running in OCCNE cluster under namespace occne-infra.
cmServiceNodePort	Custom node port for CM service	No	0	When not specified, kubernetes assigns a random port.
bsfDiamGatewayNodePort	Custom node port for Diameter Gateway service	No	0	When not specified, kubernetes assigns a random port.

Table 5-2 Mandatory Fixed Value Parameters

Name	Value
envManageNF	'BSF'
envSystemName	'BSF'
imageServiceDetector	readiness-detector:1.4.0

Core Services

Table 5-3 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Notes
deploymentBsfManagementService.envMysqlDatabase	Name of the database for BSF Management Service	No	ocpm_bsf	Specify the value of deployment BsfManagement Service. envMysqlDatabase. If no value is specified for deployment BsfManagement Service. envMysqlDatabase then assign "ocpm_bsf".

Common Services

Table 5-4 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Notes
deploymentBsfManagementService.envMysqlDatabaseSessionViewer	Name of the database for Session viewer. (BSF Management Service database name)	Yes	Not applicable	
deploymentPcfConfig.envMysqlDatabase	Name of the database for Config Server service	No	ocpm_config_server	
deploymentNrfClientservice.envNfNamespace	Namespace of BSF	Yes	Not applicable	Specify the namespace of this BSF instance.
configmapApplicationConfig.nrfHost	IP Address or hostname(Fqdn) of NRF	Yes	Not applicable	
configmapApplicationConfig.nrfServicePort	Service port of NRF	Yes	Not applicable	

Table 5-5 Mandatory Fixed Value Parameters

Name	Value
deploymentNrfClientservice.envNfType	'bsf'
deploymentNrfClientservice.envConsumeSvcName	'bsf-management-service:nbsf-management'
configmapApplicationConfig.nrfClientType	'BSF'

API Gateway Service

Table 5-6 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Notes
fullnameOverride	Unique name for Api Gateway deployment	Yes	Not applicable	It is recommended to add a prefix to "api-gateway", the prefix shall uniquely identify the instance of BSF. For example if the namespace of the BSF is "ocbsf-site1" then assign the name as "ocbsf-site1-api-gateway".

Table 5-6 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Notes
image.registry	Name of the Docker registry which hosts BSF api-gateway docker image.	Yes	Not applicable	Same value as **global.dockerRegistry**
service.type	Specify "API Gateway Service" service type as "NodePort" or "LoadBalancer". See Deployment Service Type Selection.	No	NodePort	
configServiceNetworkNameEnabled	To assign a load balancer (METALLB) IP,set **configServiceNetworkNameEnabled** to **true**	No	False	
configServiceNetworkName	To configure METALLB address pool	No	"metallb.universe.tf/address-pool: oam"	Default value is for METALLB, as per customer metallb setting, please select an appropriate pool name to replace "oam" with actual address pool.

6
Uninstalling Binding Support Function

When you uninstall a Helm chart from your BSF deployment, it removes only the Kubernetes objects that it created during installation.

To uninstall, enter this command:

```
helm delete BSF_NAME
```

where *BSF_NAME* is the release name used by helm command.

Helm keeps a record of its releases, so you can still re-activate the release after you uninstall it.

To completely remove the release from the cluster, add the --purge option to the command:

```
helm delete --purge BSF_NAME
```

For example, to completely remove a release named "ocbsf", you'd enter this command:

```
helm delete --purge ocbsf
```

Deleting Kubernetes namespace

To delete kubernetes namespace, enter this command:

```
kubectl delete namespace BSF_NAMESPACE
```

where *BSF_NAMESPACE* is the deployment namespace used by helm command.

For example, to delete a kubernetes namespace named "ocbsf", you'd enter this command:

```
kubectl delete namespace ocbsf
```

A

Deployment Service Type Selection

Service Type	Description
ClusterIP	Exposes the service on a cluster-internal IP. Specifying this value makes the service only reachable from within the cluster. This is the default ServiceType.
NodePort	Exposes the service on each worker node's IP (public IP address) at a static port (the NodePort). A ClusterIP service, to which the NodePort service will route, is automatically created. You'll be able to contact the NodePort service, from outside the cluster, by requesting *NodeIP*:*NodePort* . Most BSF service use NodePort to deploy in this release.
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. For CM Service, API gateway, Diameter Gateway service, it's recommended to use LoadBalancer type. Given that the CNE already integrated with a load balancer (METALLB, for OCCNE deployed on baremetal).