Oracle® Communications

Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide

Release 24.2.2
F98391-03
September 2025

Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide, Release 24.2.2

F98391-03

Contents

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.

Acronyms

The following table lists the acronyms and the terminologies used in the document:

Table Acronyms and Terminologies

Acronym	Description
3GPP	3rd Generation Partnership Project
5GC	5G Core Network
5GS	5G System
AF	Application Function
API	Application Programming Interface
CAPIF	Oracle Communications Common API Framework
CNC	Oracle Communications Cloud Native Core
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
ECUR	Event Charging with Unit Reservation
FQDN	Fully Qualified Domain Name
GPSI	Generic Public Subscription Identifier
GMLC	Gateway Mobile Location Center
HA	High Availability
IEC	Immediate Event Charging
IGW	Internet Gateway
IMSI	International Mobile Subscriber Identity
K8s	Kubernetes
ME	Monitoring Events
MSISDN	Mobile Station International Subscriber Directory Number
NEF	Oracle Communications Cloud Native Core, Network Exposure Function
NF	Network Function
NRF	Oracle Communications Cloud Native Core, Network Repository Function
NSSF	Oracle Communications Cloud Native Core, Network Slice Selection Function
OAM	Operations, Administration, and Maintenance
OCI	Oracle Cloud Infrastructure
OKE	Oracle Kubernetes Engine
PDB	Pod Disruption Budget
QoS	Quality of Service
SBA	Service Based Architecture
SBI	Service Based Interface
SCEF	Service Capability Exposure Function
S-NSSAI	Single Network Slice Selection Assistance Information
SUPI	Subscription Permanent Identifier
TPS	Traffic Per Second
UDM	Unified Data Management

Table (Cont.) Acronyms and Terminologies

Acronym	Description
URI	Uniform Resource Identifier

What's New in This Guide

This section introduces the documentation updates for Release 24.2.x.

Release 24.2.2 - F98391-03, September 2025

- **General Updates:**
  - Updated the version to 24.2.2 in the entire document.
  - Updated the image tag details in the Pushing the Images to Customer Docker Registry and Pushing the Images to Customer Docker Registry.
- **Upgrade, Rollback, and Uninstall Updates**:
  - Updated the supported upgrade paths in the Supported Upgrade Paths section for CAPIF.
  - Updated the supported rollback paths in the Supported Rollback Paths section for CAPIF.
  - Updated the supported upgrade paths in the Supported Upgrade Paths section for NEF.
  - Updated the supported rollback paths in the Supported Rollback Paths section for NEF.

Release 24.2.1 - F98391-02, October 2024

- **General Updates:**
  - Updated the version to 24.2.1 in the entire document.
  - Updated the image tag details in the Pushing the Images to Customer Docker Registry and Pushing the Images to Customer Docker Registry.
- **Upgrade, Rollback, and Uninstall Updates**:
  - Updated the supported upgrade paths in the Supported Upgrade Paths section for CAPIF.
  - Updated the supported rollback paths in the Supported Rollback Paths section for CAPIF.
  - Updated the supported upgrade paths in the Supported Upgrade Paths section for NEF.
  - Updated the supported rollback paths in the Supported Rollback Paths section for NEF.

Release 24.2.0-beta - F98391-01, July 2024

The following changes are made in this release:

- **General Updates:**
  - Updated the version to 24.2.0-beta in the entire document.
  - Replaced "blacklist" with "blocklist" in the entire document to comply with inclusive language.
  - Updated the image tag details in the Pushing the Images to Customer Docker Registry and Pushing the Images to Customer Docker Registry.

- **Upgrade, Rollback, and Uninstall Updates**:
  - Updated the supported upgrade paths in the Supported Upgrade Paths section for CAPIF.
  - Updated the supported rollback paths in the Supported Rollback Paths section for CAPIF.
  - Updated the supported upgrade paths in the Supported Upgrade Paths section for NEF.
  - Updated the supported rollback paths in the Supported Rollback Paths section for NEF.
- Updated software versions in Software Requirements to provide the latest software versions used.
- The following changes are made as part of the Deployment in OCI using OCI Adaptor feature:
  - Added the acronym, OCI and OKE, in the Acronyms section to provide abbreviations of these acronyms.
  - Added the *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide* document in the References section.
  - Updated the Introduction section with supported deployment platforms.
  - Added preinstalled software for OCI in the Software Requirements section.
  - Added information about OCI in the OCI Requirements section.
  - Added the following sections with information related to pushing images to the OCI registry:
    * Pushing the NEF Images to OCI Registry
    * Pushing the CAPIF Images to OCI Registry
  - Added the following files in the Customizing NEF section:
    * `ocnef_oci_alertrules_24.2.0.zip`
    * `ocnef_oci_metric_dashboard_24.2.0.zip`
- Added a note on updating configurations only through Rest API in Configurable Parameters of NEF and Configurable Parameters of CAPIF sections.

1 Introduction

This guide describes how to install or upgrade Oracle Communications Cloud Native Core, Network Exposure Function (NEF) and Oracle Communications Common API Framework (CAPIF) in a cloud native environment and Oracle Cloud Infrastructure (OCI). It also includes information on performing fault recovery for NEF.

Note

- This guide covers the installation instructions when Podman is the container platform with Helm as the Packaging Manager. For any other container platform, the operator must use the commands based on to their deployed container runtime environment.
- `kubectl` commands can vary based on the platform deployment. Replace `kubectl` with Kubernetes environment-specific command line tool to configure Kubernetes resources through kube-api server. The instructions provided in this document are as per the CNE version of kube-api server.

Caution

User, computer and applications, and character encoding settings may cause an issue when copy-pasting commands or any content from PDF. PDF reader version also affects the copy-pasting functionality. It is recommended to verify the pasted content especially when the hyphens or any special characters are part of the copied content.

1.1 Overview

NEF is a key component of the 5G Service Based Architecture (SBA). NEF provides a platform to securely expose the network services and capabilities offered by the 5G Network Functions (NFs) to either external third-party applications or the internal Application Functions (AFs). As an interface between 5G core network and different application functions, NEF enables the operators to customize their network and provide innovative services to the end users.

Note

The performance and capacity of the NEF system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.

NEF supports the following functions:

- Facilitates robust and secure exposure of network services, such as voice, data connectivity, charging, subscriber data, IoT, and so on to trusted third-party applications or AFs
- Provides programmable environment access of 5G network to operators

- Provides 3GPP defined API exposure gateway to build new services
- Enables AF to securely provide information to 3GPP network to authenticate, authorize, and assist in throttling the AF
- Translates the information received from AF to the one sent to internal 3GPP NFs, and vice-versa
- Provides support to expose the information collected from other 3GPP NFs to the AF

NEF interacts with different application functions and the 5G core network to support the above functions. The application consists of the following components running on seperate namespaces in the cloud native environment:

> **Note**
>
> CAPIF and NEF can be installed in different cluster also, but the cluster has to be reachable.

- **Common API Framework (CAPIF):** The service interfaces between NEF and the external third-party applications or internal Application Functions (AFs).
  CAPIF is a 3GPP defined secured framework to expose network service interfaces. It enables the API invokers (external applications) to discover and communicate with service APIs of the API provider (NEF). This framework manages API security, logging of events, auditing capability, multiple service exposure, policy based routing, dynamic routing of information, and so on. CAPIF consists of the following services:
  - AF Manager
  - API Manager
  - Event Manager Service
  - External Ingress Gateway
  - External Egress Gateway
  - Network Ingress Gateway
  - Network Egress Gateway
- **Network Exposure Function (NEF):** The core component that runs the business logic of NEF. It consists of various services that interact with the CAPIF and perform the core functionality of NEF. NEF provides the following services:
- 5GC Agent
- APD Manager
- API Router
- CAPIF Core Function (CCF) Client
- Expiry Auditor Service
- Monitoring Events
- QoS Service
- External Ingress Gateway
- External Egress Gateway
- FiveGC Ingress Gateway
- FiveGC Egress Gateway

- Traffic Influence
- Device Trigger
- Pool Manager
- MSISDN-Less MO-SMS

For more information about the services, see *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

Installation and Upgrade Overview

NEF installation and upgrade procedures comprise of different tasks including CAPIF and NEF components. Perform the installation and upgrade tasks in the same sequence as outlined in the following table:

Table 1-1 Installation and Upgrade Overview

Task	Sub task	Reference Link
Installation Sequence		
Set up the installation environment	Prerequisite	Prerequisites
	Software Requirements	Software Requirements
	Environment Setup Requirements	Environment Setup Requirements
	Resource Requirements	Resource Requirements
Download installation package	Downlaod Package	Installation Package Download
Perform CAPIF preinstallation tasks	Create CAPIF Namespace	Verifying and Creating CAPIF Namespace
	Configure CAPIF databases	Configuring Database, Creating Users, and Granting Permissions
	Configure Secrets	Configuring Kubernetes Secret for Accessing CAPIF Database
	Configure Security	Configuring Secrets for Enabling HTTPS
Perform CAPIF installation tasks	Push images to docker registry	Pushing the Images to Customer Docker Registry
	Customize CAPIF Custom Values YAML file	Customizing CAPIF
	Install CAPIF	Installing CAPIF Package
	Perform Helm Test	Performing Helm Test
Verify CAPIF installation	Verifying CAPIF Installation	Verifying CAPIF Installation
Perform NEF preinstallation tasks	Create NEF Namespace	Verifying and Creating NEF Namespace
	Configure NEF databases	Configuring Database, Creating Users, and Granting Permissions

Table 1-1 (Cont.) Installation and Upgrade Overview

Task	Sub task	Reference Link
	Configure Secrets	**Configuring Kubernetes Secret for Accessing NEF Database**
	Configure Security	Configuring Secrets for Enabling HTTPS
Perform NEF installation tasks	Push images to docker registry	Pushing the Images to Customer Docker Registry
	Customize NEF Custom Values YAML file	Customizing NEF
	Install NEF	Installing NEF Package
Verify NEF installation	Verifying NEF Installation	Verifying NEF Installation
Upgrade Sequence		
Perform CAPIF upgrade tasks	Preupgrade tasks	Preupgrade Tasks
	Upgrade CAPIF	Upgrade Tasks
Perform NEF upgrade tasks	Upgrade NEF	Upgrade Tasks
Rollback Sequence		
Perform NEF rollback	Rollback NEF	Rolling Back NEF
Perform CAPIF rollback	Rollback CAPIF	Rolling Back CAPIF

1.2 References

Refer to the following documents while deploying NEF:

- *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*
- *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*
- *Oracle Communications Cloud Native Core, Network Exposure Function Compatibility Matrix*
- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*

1.3 Oracle Error Correction Policy

The table below outlines the key details for the current and past releases, their General Availability (GA) dates, the latest patch versions, and the end dates for the Error Correction Grace Period.

Table 1-2 Error Correction Policy

Release Number	General Availability (GA) Date	Latest Patch Version	Error Correction Grace Period End Date
3.24.2	July 2024	24.2.2	July 2025

Table 1-2 (Cont.) Error Correction Policy

Release Number	General Availability (GA) Date	Latest Patch Version	Error Correction Grace Period End Date
3.24.1	April 2024	24.1.0	April 2025
3.23.4	December 2023	23.4.3	December 2024
2.23.3	September 2023	23.3.2	September 2024

> **Note**
>
> For a release, Sev1 and Critical Patch Unit (CPU) patches are supported for 12 months. For more information, see the Oracle Communications Cloud Native Core and Network Analytics Error Correction Policy.

1.4 Open Source Support Policies

Oracle Communications Cloud Native Core uses open source technology governed by the Oracle Open Source Support Policies. For more information, see the Oracle Open Source Support Policies.

2
Prerequisites

Before installing and configuring NEF, ensure that the following prerequisites are met:

- Software Requirements
- Environment Setup Requirements
- Resource Requirements

The NEF installation must be performed in the following order:

1. Installing CAPIF
2. Installing NEF

2.1 Software Requirements

This section lists the software that must be installed before installing NEF:

Table 2-1 Preinstalled Software

Software	Version
Kubernetes	1.29.1, 1.28.6, 1.27.x
Helm	3.13.2
Podman	3.3.1, 3.2.3, 2.2.1

> **Note**
>
> - NEF 24.2.x supports CNE 24.1.x, 23.4.x, and 23.3.x.
> - NEF 24.2.x supports OKE managed clusters on OCI.

To check the versions of the preinstalled software in the cloud native environment, run the following commands:

```
kubectl version
```

```
helm version
```

```
podman version
```

The following software are available if NEF is deployed in CNE. If you are deploying NEF in any other cloud native environment, these additional software must be installed before installing the NEF.

To check the installed software, run the following command:

```
helm ls -A
```

The list of additional software items, along with the supported versions and usage, is provided in the following table:

Table 2-2 Additional Software

Software	App Version	Required For
OpenSearch	2.11.0	Logging
OpenSearch Dashboard	2.11.0	Logging
logs	3.1.0	Logging
Kyverno	1.9.0	Logging
Grafana	9.5.3	Metrics and KPIs
Prometheus	2.51.1	Metrics and Alerts
Prometheus Operator	0.72.0	Metrics
MetalLb	0.14.4	External IP
metrics-server	0.6.1	Metrics
tracer	1.21.0	Tracing
Jaeger	1.52.0	Tracing
snmp-notifier	1.4.0	Alerts

> **Note**
>
> On OCI, the above mentioned software are not required because OCI observability and management service is used for logging, metrics, alerts, and KPIs. For more information, see *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*.

2.2 Environment Setup Requirements

This section describes the environment setup requirements for installing NEF.

2.2.1 Network Access Requirement

The Kubernetes cluster hosts must have network access to the following repositories:

- **Local helm repository** – It contains the CAPIF and NEF helm charts.
  To check if the Kubernetes cluster hosts can access the local helm repository, run the following command:

  ```
  helm repo update
  ```

- **Local docker image repository** – It contains the CAPIF and NEF docker images.

To check if the Kubernetes cluster hosts can access the local docker image repository, pull any image with an image-tag, using the following commands:

```
docker pull <Docker-repo>/<image-name>:<image-tag>
```

```
podman pull <Docker-repo>/<image-name>:<image-tag>
```

where:

<Docker-repo> is the IP address or host name of the Docker repository.

<Podman-repo> is the IP address or host name of the Podman repository.

<image-name> is the Docker image name.

<image-tag> is the tag assigned to the Docker image used for the CAPIF and NEF pods.

For example:

```
docker pull CUSTOMER_REPO/ocnef_monitoring_events:24.2.2
```

```
podman pull docker-repo/ocnef_monitoring_events:24.2.2
```

> **Note**
>
> Run the `kubectl` and `helm` commands on a system based on the deployment infrastructure. For instance, it can be run on a client machine such as VM, server, local desktop, and so on.

2.2.2 Client Machine Requirement

This section describes the requirements for client machine, that is, the machine used by the user to run deployment commands.

The client machine should have:

- Helm repository configured.
- network access to the Helm repository and Docker image repository.
- network access to the Kubernetes cluster.
- required environment settings to run the `kubectl`, `docker`, and `podman` commands. The environment must have privileges to create a namespace in the Kubernetes cluster.
- Helm client installed with the push plugin. Configure the environment in such a manner that the `helm install` command deploys the software in the Kubernetes cluster.

2.2.3 Server or Space Requirement

For information about server or space requirements, see the *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide* .

2.2.4 CNE Requirement

This section is applicable only if you are installing NEF on Cloud Native Environment (CNE).

NEF supports CNE 24.1.x, 23.4.x, 23.3.x, and 23.2.x.

To check the CNE version, run the following command:

```
echo $OCCNE_VERSION
```

> **Note**
>
> For more information about CNE, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

2.2.5 OCI Requirements

NEF can be deployed in OCI. While deploying NEF in OCI, the user must use the Operator instance/VM instead of Bastion Host.

For more information about OCI Adaptor, see *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*.

2.2.6 cnDBTier Requirement

NEF supports cnDBTier 24.1.x, 23.4.x, 23.3.x, and 23.2.x in a virtual CNE (vCNE) environment. cnDBTier must be up and active in case of containerized CNE. For more information on installation procedure, see the *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

To install NEF or CAPIF with recommended cnDBTier resources, install cnDBTier using the `ocnef_dbtier_24.2.2_custom_values_24.2.2.yaml` file provided in the `ocnef_custom_configTemplates_24.2.2` file. For information about the steps to download `ocnef_custom_configTemplates_24.2.2` file, see Customizing NEF or Customizing CAPIF.

> **Note**
>
> There are no customizations required in the `ocnef_dbtier_24.2.2_custom_values_24.2.2.yaml` file for installing NEF 24.2.2.

2.2.6.1 cnDBtier Customization Parameters

Table 2-3 cnDBtier Customization Parameters

Parameter Name	Recommended Value	Added/Modified in Release
db-monitor-svc.resources.limits.cpu	4	24.2.2
db-monitor-svc.resources.limits.memory	2Gi	24.2.2
db-monitor-svc.resources.limits.ephemeral-storage	1Gi	24.2.2

Table 2-3 (Cont.) cnDBtier Customization Parameters

Parameter Name	Recommended Value	Added/Modified in Release
db-monitor-svc.resources.requests.cpu	4	24.2.2
db-monitor-svc.resources.requests.memory	2Gi	24.2.2
db-monitor-svc.resources.requests.ephemeral-storage	1Gi	24.2.2
ndb.resources.limits.cpu	6	24.2.2
ndb.resources.limits.memory	18Gi	24.2.2
ndb.resources.limits.ephemeral-storage	1Gi	24.2.2
ndb.resources.requests.cpu	6	24.2.2
ndb.resources.requests.memory	16Gi	24.2.2
ndb.resources.requests.ephemeral-storage	90Mi	24.2.2
additionalndbconfigurations.mysqld.ndb_allow_copying_alter_table	OFF	24.2.2
api.resources.limits.cpu	4	24.2.2
api.resources.limits.memory	10Gi	24.2.2
api.resources.limits.ephemeral-storage	1Gi	24.2.2
api.resources.requests.cpu	4	24.2.2
api.resources.requests.memory	10Gi	24.2.2
api.resources.requests.ephemeral-storage	90Mi	24.2.2
api.ndbapp.resources.limits.cpu	5	24.2.2
api.ndbapp.resources.limits.memory	10Gi	24.2.2
api.ndbapp.resources.limits.ephemeral-storage	1Gi	24.2.2
api.ndbapp.resources.requests.cpu	5	24.2.2
api.ndbapp.resources.requests.memory	10Gi	24.2.2
api.ndbapp.resources.requests.ephemeral-storage	90Mi	24.2.2

2.3 Resource Requirements

This section lists the resource requirements to install and run NEF and CAPIF.

> **Note**
>
> The performance and capacity of the NEF system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.

2.3.1 Services

2.3.1.1 NEF Services

The following table lists the resource requirement for NEF Services:

Table 2-4 NEF Services

Service Name	POD		CPU		Memory	
	Default POD Count	Maximum POD count with scaling	Min	Max	Min	Max
APD Manager	2	2	4	4	4Gi	4Gi
API Router	2	2	4	4	4Gi	4Gi
External Egress Gateway	2	5	4	4	4Gi	4Gi
External Ingress Gateway	2	5	4	4	4Gi	4Gi
5GC Egress Gateway	2	5	4	4	4Gi	4Gi
5GC Ingress Gateway	2	5	4	4	4Gi	4Gi
ME Service	2	12	4	4	4Gi	4Gi
QoS Service	2	12	4	4	4Gi	4Gi
5GC Agent	2	12	4	4	4Gi	4Gi
CCF Client	2	3	2	2	2Gi	2Gi
Expiry Auditor	2	3	4	4	4Gi	4Gi
Perf-Info	1	1	1	1	1Gi	1Gi
App-Info	1	1	1	1	1Gi	1Gi
Config-Server	1	1	1	1	1Gi	1Gi
NRF Client	1	1	1	1	1Gi	1Gi
Traffic Influence	2	12	4	4	4Gi	4Gi
Diameter Gateway	2	2	4	4	4Gi	4Gi
Device Trigger	1	1	1	1	1Gi	1Gi
Pool Manager	1	1	4	4	4Gi	4Gi
MSISDNless MO SMS	1	12	4	4	4Gi	4Gi
Console Data Service	1	12	4	4	4Gi	4Gi

2.3.1.2 CAPIF Services

The following table lists the resource requirement for CAPIF Services:

Table 2-5 CAPIF Services

Service Name	POD		CPU		Memory	
	Default POD Count	Maximum POD count with scaling	Min	Max	Min	Max
API Manager	2	3	4	4	4Gi	4Gi
AF Manager	2	3	4	4	4Gi	4Gi
Event Manager	2	2	4	4	4Gi	4Gi

Table 2-5 (Cont.) CAPIF Services

Service Name	POD		CPU		Memory	
	Default POD Count	Maximum POD count with scaling	Min	Max	Min	Max
External Ingress Gateway	2	5	4	4	4Gi	4Gi
External Egress Gateway	2	5	2	2	4Gi	4Gi
Network Ingress Gateway	2	5	4	4	4Gi	4Gi
Network Egress Gateway	2	5	4	4	4Gi	4Gi

2.3.2 Debug Tool Container

The Debug Tool provides third-party troubleshooting tools for debugging the runtime issues in lab environment. If Debug Tool Container injection is enabled during NEF deployment or upgrade, this container is injected to each NEF pod (or selected pod, depending on the option chosen during deployment or upgrade). These containers stay till pod or deployment exist. For more information about Debug Tool, see *Oracle Communication Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

Table 2-6 NEF - Debug Tool Container

Service Name	CPU		Memory		POD		Ephemeral Storage	
	Min	Max	Min	Max	Min	Max	Min	Max
APD Manager	0.5	0.5	4Gi	4Gi	2	2	512Mi	512Mi
API Router	0.5	0.5	4Gi	4Gi	2	2	512Mi	512Mi
External Egress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi
External Ingress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi
5GC Egress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi
5GC Ingress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi
ME Service	0.5	0.5	4Gi	4Gi	2	12	512Mi	512Mi
QoS Service	0.5	0.5	4Gi	4Gi	2	12	512Mi	512Mi
5GC Agent	0.5	0.5	4Gi	4Gi	2	12	512Mi	512Mi
CCF Client	0.5	0.5	4Gi	4Gi	2	3	512Mi	512Mi
Expiry Auditor	0.5	0.5	4Gi	4Gi	2	3	512Mi	512Mi
Perf-Info	0.5	0.5	4Gi	4Gi	1	1	512Mi	512Mi

Table 2-6 (Cont.) NEF - Debug Tool Container

Service Name	CPU		Memory		POD		Ephemeral Storage	
	Min	Max	Min	Max	Min	Max	Min	Max
App-Info	0.5	0.5	4Gi	4Gi	1	1	512Mi	512Mi
Config-Server	0.5	0.5	4Gi	4Gi	1	1	512Mi	512Mi
NRF Client	0.5	0.5	4Gi	4Gi	1	1	512Mi	512Mi
Traffic Influence	0.5	0.5	4Gi	4Gi	2	12	512Mi	512Mi
Diameter Gateway	0.5	0.5	4Gi	4Gi	2	2	512Mi	512Mi
Device Trigger	0.5	0.5	1Gi	1Gi	1	1	512Mi	512Mi
Pool Manager	1	1	1Gi	1Gi	2	12	512Mi	512Mi
Console Data Service	1	1	1Gi	1Gi	1	12	512Mi	512Mi
MSISDNLess MO SMS	1	1	1Gi	1Gi	1	12	512Mi	512Mi

> **Note**
>
> The debug container resources are optional. If you are using the debug container, then the **CPU Request and Limit for Debug Container** and **Memory Request and Limit for Debug Container** values must be included to the minimum and maximum resources for NEF services.

Table 2-7 CAPIF - Debug Tool Container

Service Name	CPU		Memory		POD		Ephemeral Storage	
	Min	Max	Min	Max	Default POD Count	Maximum POD count with scaling	Min	Max
API Manager	0.5	0.5	4Gi	4Gi	2	3	512Mi	512Mi
AF Manager	0.5	0.5	4Gi	4Gi	2	3	512Mi	512Mi
Event Manager	0.5	0.5	4Gi	4Gi	2	2	512Mi	512Mi
External Ingress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi
External Egress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi

Table 2-7 (Cont.) CAPIF - Debug Tool Container

Service Name	CPU		Memory		POD		Ephemeral Storage	
	Min	Max	Min	Max	Default POD Count	Maximum POD count with scaling	Min	Max
Network Ingress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi
Network Egress Gateway	0.5	0.5	4Gi	4Gi	2	5	512Mi	512Mi

> **Note**
>
> The debug container resources are optional. If you are using the debug container, then the **CPU Request and Limit for Debug Container** and **Memory Request and Limit for Debug Container** values must be included to the minimum and maximum resources for CAPIF services.

2.3.3 Upgrade

2.3.3.1 NEF Upgrade

The following table lists the resource requirement for NEF Upgrade:

Table 2-8 NEF Upgrade

Service Name	POD		CPU		Memory	
	Default POD Count	Maximum POD count with scaling	Min	Max	Min	Max
APD Manager	2	2	4	4	4Gi	4Gi
API Router	2	2	4	4	4Gi	4Gi
External Egress Gateway	2	5	4	4	4Gi	4Gi
External Ingress Gateway	2	5	4	4	4Gi	4Gi
5GC Egress Gateway	2	5	4	4	4Gi	4Gi
5GC Ingress Gateway	2	5	4	4	4Gi	4Gi
ME Service	2	12	4	4	4Gi	4Gi
QoS Service	2	12	4	4	4Gi	4Gi
5GC Agent	2	12	4	4	4Gi	4Gi
CCF Client	2	3	2	2	2Gi	2Gi
Expiry Auditor	2	3	4	4	4Gi	4Gi

Table 2-8 (Cont.) NEF Upgrade

Service Name	POD		CPU		Memory	
	Default POD Count	Maximum POD count with scaling	Min	Max	Min	Max
Perf-Info	1	1	1	1	1Gi	1Gi
App-Info	1	1	1	1	1Gi	1Gi
Config-Server	1	1	1	1	1Gi	1Gi
NRF Client	1	1	1	1	1Gi	1Gi
Traffic Influence	2	12	4	4	4Gi	4Gi
Diameter Gateway	2	2	4	4	4Gi	4Gi
Device Trigger	1	1	1	1	1Gi	1Gi
Pool Manager	1	1	4	4	4Gi	4Gi
MSISDNless MO SMS	1	12	4	4	4Gi	4Gi
Console Data Service	1	12	4	4	4Gi	4Gi

2.3.3.2 CAPIF Upgrade

The following table lists the resource requirement for CAPIF Upgrade:

Table 2-9 CAPIF Upgrade

Service Name	POD		CPU		Memory	
	Default POD Count	Maximum POD count with scaling	Min	Max	Min	Max
API Manager	2	3	4	4	4Gi	4Gi
AF Manager	2	3	4	4	4Gi	4Gi
Event Manager	2	2	4	4	4Gi	4Gi
External Ingress Gateway	2	5	4	4	4Gi	4Gi
External Egress Gateway	2	5	2	2	4Gi	4Gi
Network Ingress Gateway	2	5	4	4	4Gi	4Gi
Network Egress Gateway	2	5	4	4	4Gi	4Gi

2.3.4 Common Services Container

Following is the resource requirement for Common Services Container.

Table 2-10 Common Services Container

Container Name	CPU Request and Limit Per Container	Memory (GB) Request and Limit Per Container	Kubernetes Init Container (Job)
init-service	1	1	Y
update-service	1	1	N
common_config_hook	1	1	N

- **Update Container service:** Ingress or Egress Gateway services use this container service to periodically refresh CAPIF Private Key or Certificate and CA Root Certificate for TLS.
- **Init Container service:** Ingress or Egress Gateway services use this container to get CAPIF Private Key or Certificate and CA Root Certificate for TLS during start up.

2.3.5 Hooks

Following is the resource requirement for Hooks for Monitoring Events microservice.

Table 2-11 Hooks for Monitoring Events Microservice

Hook Name	CPU	Memory
<helm-release-name>-monitoringevents-pre-install	1	1Gi
<helm-release-name>-monitoringevents-post-install	1	1Gi
<helm-release-name>-monitoringevents-pre-upgrade	1	1Gi
<helm-release-name>-monitoringevents-post-upgrade	1	1Gi
<helm-release-name>-monitoringevents-pre-rollback	1	1Gi
<helm-release-name>-monitoringevents-post-rollback	1	1Gi
<helm-release-name>-monitoringevents-pre-delete	1	1Gi
<helm-release-name>-monitoringevents-post-delete	1	1Gi

Note

- <helm-release-name> is the Helm release name. For example, if Helm release name is "ocnef", then monitoringevents microservice name will be "ocnef-monitoringevents"
- The above table provides the hooks for `monitoringevents` microservice. Similar hooks are applicable for all the following microservices:
  - NEF microservices:
    - apdmanager
    - ocnef-expiry-auditor
    - nrfclient
    - aef-apirouter
    - app-info
    - perf-info
    - config-server
    - monitoringevents
    - qualityofservice
    - trafficinfluence
    - ocnef-ccfclient
    - devicetrigger
    - poolmanager
    - fivegcagent
    - ingress-gateway
    - egress-gateway
    - ocnef-diam-gateway
    - msisdnless_mo_sms
    - console_data_service
  - CAPIF microservices:
    - apimgr
    - afmgr
    - eventmgr
    - ingress-gateway
    - egress-gateway
    - console_data_service

3
Installation Package Download

To download the NEF package from My Oracle Support, perform the following steps:

1. Log in to My Oracle Support using the appropriate credentials.
2. Select **Patches & Updates** tab.
3. In **Patch Search** console, select the **Product or Family (Advanced)** option.
4. Enter `Oracle Communications Cloud Native Core - 5G` in **Product** field.
5. From the **Release** drop-down, select **"Oracle Communications Cloud Native Core Network Exposure Function <release_number>"**.
   Where, `<release_number>` indicates the required release version of NEF.
6. Click **Search**.
   The Patch Advanced Search Results list appears.
7. Select the required patch from the list.
   The Patch Details window appears.
8. Click **Download**.
   File Download window appears.
9. Click **<p********_<release_number>_Tekelec>.zip** to download the release package.

4
CAPIF Installation and Upgrade

This chapter describes how to install, customize, upgrade, and uninstall Common API Framework (CAPIF) on Oracle Communications Cloud Native Environment (CNE).

4.1 Installing CAPIF

This section provides information about installing Common API Framework (CAPIF) in a cloud native environment.

> **Note**
>
> CAPIF supports the independent deployment with its microservices. For information about the prerequisites to install CAPIF, see Prerequisites chapter and to know how to upgrade CAPIF, see Upgrading CAPIF section.

4.1.1 Preinstallation

Before installing CAPIF, perform the tasks described in this section.

> **Note**
>
> CAPIF supports fresh installation, and it can also be upgraded from 22.3.x. For more information on how to upgrade CAPIF, see Upgrading CAPIF section.

4.1.1.1 Verifying and Creating CAPIF Namespace

This section explains how to verify or create a new namespace in the system.

> **Note**
>
> This is a mandatory procedure, run this before proceeding further with the installation procedure. The namespace created or verified in this procedure is an input for the next procedures.

1. Run the following command to verify if the required namespace already exists in the system:

   ```
   kubectl get namespaces
   ```

   In the output of the above command, if the namespace exists, continue with Configuring Database, Creating Users, and Granting Permissions.

2. If the required namespace is unavailable, create the namespace using the following command:

   ```
   kubectl create namespace <required namespace>
   ```

   For example:
   ```
   $ kubectl create namespace occapif-namespace
   ```

   > **Note**
   >
   > This is an optional step. Skip this step if the required namespace already exists.

Naming Convention for Namespaces

The namespace should:

- start and end with an alphanumeric character.
- contain 63 characters or less.
- contain only alphanumeric characters or '-'.

> **Note**
>
> It is recommended to avoid using prefix `kube-` when creating namespace as this prefix is reserved for Kubernetes system namespaces.

4.1.1.2 Creating Service Account, Role, and RoleBinding

This section describes the procedure to create a service account, role, and role binding resources. The Secret(s) can be under same namespace where CAPIF is getting deployed (recommended), or you can select to use different namespaces for different secret(s). If all the secret(s) are under the same namespace as CAPIF, then you can bind the Kubernetes Role with the given ServiceAccount. Otherwise, it is required to bind the ClusterRole with the given ServiceAccount.

1. Create an CAPIF resource file:

   ```
   vi <occapif-resource-file>
   ```

   For example:

   ```
   vi occapif-resource-template.yaml
   ```

2. Update the `occapif-resource-template.yaml` with release specific information:

> **Note**
>
> Update <Helm-release> and <namespace> with its respective CAPIF namespace and CAPIF helm release name.

```
## Sample template start#
apiVersion: v1
kind: ServiceAccount
metadata:
  name: <helm-release>-occapif-serviceaccount
  namespace: <namespace>
---

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: <helm-release>-occapif-role
  namespace: <namespace>
rules:
- apiGroups:
  - "" # "" indicates the core API group
  resources:
  - services
  - configmaps
  - pods
  - secrets
  - endpoints
  verbs:
  - get
  - watch
  - list
---

apiVersion: rbac.authorization.k8s.io/v1beta1
kind: RoleBinding
metadata:
  name: <helm-release>-occapif-rolebinding
  namespace: <namespace>
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: <helm-release>-occapif-role
subjects:
- kind: ServiceAccount
  name: default
  namespace: <namespace>
## Sample template end#
```

3. Run the following command to create serviceaccount, role and role binding:

```
kubectl -n <occapif-namespace> create -f occapif-resource-template.yaml
```

For example:

```
kubectl -n occapif create -f occapif-resource-template.yaml
```

4. Update the `serviceAccountName` parameter in the `oc-capif-24.2.2-custom-values.yaml` file with the value updated in `name` field under `kind: ServiceAccount`. For more information about `serviceAccountName` parameter, see the Global Parameters section.

> **Note**
>
> The service account name configured in this section must be used as the value for the `serviceAccountName` parameter during the customization using the CAPIF Custom Values YAML file. For more information about the parameter, see Global Parameters.

4.1.1.3 Configuring Database, Creating Users, and Granting Permissions

This section explains how database administrators can create users and database in a single and multisite deployment.

CAPIF microservices use MySQL database to store the configuration and run time data.

CAPIF requires the database administrator to create a user in MySQL database and provide the necessary permissions to access the databases. Before installing CAPIF, it is required to create the MySQL user and databases.

> **Note**
>
> Before running the procedure for georedundant sites, ensure that the cnDBTier for georedundant sites is already up and replication channels are enabled. While performing a fresh installation, if CAPIF release is already deployed, purge the deployment and remove the database and users that were used for the previous deployment. For uninstallation procedure, see Uninstalling CAPIF.

CAPIF Users

There are two types of CAPIF database users with a different set of permissions:

1. **CAPIF privileged user:** This user has a complete set of permissions. This user can create or delete the database and perform create, alter, or drop operations on the database tables for performing installation, upgrade, rollback, and delete operations.
2. **CAPIF application user:** This user has a limited set of permissions and is used by CAPIF application during service operations handling. This user can insert, update, get, and remove the records. This user cannot create, alter, and drop the database and the tables.

4.1.1.3.1 Single Site

This section explains how a database administrator can create database and users for a single CAPIF site deployment.

> **Note**
>
> New MySQL users along with their privileges must be added manually in each SQL node of the cnDBtier namespace, for CAPIF site.

1. Log in to the machine where SSH keys are stored. The machine must have permission to access the SQL nodes of NDB cluster.
2. Connect to the SQL nodes.
3. Log in to the database as a root user.
4. Create the CAPIF Release database:

   ```
   CREATE DATABASE <capif_release_database_name>;
   ```

   Example

   ```
   CREATE DATABASE occapif_releaseDb;
   ```

   > **Note**
   >
   > In case of georedundant deployment, each CAPIF site must have different Release database names.

5. Create the CAPIF Service database:

   ```
   CREATE DATABASE <capif_service_database_name>;
   ```

   Example

   ```
   CREATE DATABASE occapif_db;
   ```

   > **Note**
   >
   > In case of georedundant deployment, each CAPIF site must have different Service database names.

6. Create the CAPIF privileged user and grant permissions.
   a. Run the following command to create privileged user:

      ```
      CREATE USER '<capif privileged username>'@'%' IDENTIFIED BY '<capif
      privileged user password>';
      ```

> **Note**
>
> If MySQL is 8.0, run the following command to create a user:
>
> ```
> CREATE USER IF NOT EXISTS '<capif privileged username>@'%'
> IDENTIFIED WITH mysql_native_password BY '<capif privileged user
> password>'
> ```

where:

<capif privileged username> is the username and *<capif privileged user password>* is the password for MySQL privileged user.

b. Run the following command to grant the necessary permissions to the privileged user:

```
GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES,
CREATE TEMPORARY TABLES, DELETE, UPDATE,
EXECUTE, NDB_STORED_USER , REFERENCES ON *.* TO '<capif privileged
username>'@'%';

FLUSH PRIVILEGES;
```

Example

In the following example "occapifprivilegedusr" is used as username and "occapifprivilegedpasswd" is used as the password. All the permissions are granted to the privileged user, that is, occapifprivilegedusr.

```
CREATE USER 'occapifprivilegedusr'@'%' IDENTIFIED BY
'occapifprivilegedpasswd';



GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES,
CREATE TEMPORARY TABLES, DELETE, UPDATE,
EXECUTE, NDB_STORED_USER , REFERENCES ON *.* TO 'occapifprivilegedusr'@'%';

FLUSH PRIVILEGES;
```

7. Create the CAPIF application user and grant permissions.
   a. Run the following command to create application user:

   ```
   CREATE USER '<capif application username>'@'%' IDENTIFIED BY '<capif
   application user password>';
   ```

   where:

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

b. Run the following command to grant the necessary permissions to the privileged user:

```
GRANT SELECT, INSERT, LOCK TABLES, DELETE, UPDATE, EXECUTE, REFERENCES,
NDB_STORED_USER ON
      *.* TO '<capif application username>'@'%';
```

Example
In the following example, "occapifusr" is used as username and "occapifpasswd" is used as its password. All the necessary permissions are granted to the application user, that is, occapifusr. Here, default database names of microservices are used.

```
CREATE USER 'occapifusr'@'%' IDENTIFIED BY 'occapifpasswd';


GRANT SELECT, INSERT, LOCK TABLES, DELETE, UPDATE, EXECUTE, REFERENCES,
NDB_STORED_USER ON
      *.* TO 'occapifusr'@'%';
```

> **Note**
>
> The database name is specified in the `dbName` parameter for CAPIF services in the `oc-capif-24.2.2-custom-values.yaml` file.

8. To confirm that the privileged or application user has all the permissions, run the following command:

```
show grants for username;
```

where, *username* is the privileged or application user's username.

Example

```
show grants for occapifprivilegedusr;


show grants for occapifusr;
```

9. Exit from database and log out from MySQL node.

4.1.1.3.2 Multisite

This section explains how database administrator can create the database and users for a multisite deployment.

> **Note**
>
> - Perform the steps in Single Site for creating database.
> - NEF supports only 2-site.
> - For further information, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

For information on the Database Configurations parameters, refer to Global Parameters section.

4.1.1.4 Configuring Kubernetes Secret for Accessing CAPIF Database

This section explains how to configure Kubernetes secrets for accessing CAPIF database.

4.1.1.4.1 Creating and Updating Secret for Accessing CAPIF Privileged Database User

This section explains how to create and update Kubernetes secret for Privileged User to access the database.

1. Run the following command to create kubernetes secret:

```
$ kubectl create secret generic <privileged user secret name> --from-
literal=dbUsername=<CAPIF Privileged User Name> --from-
literal=dbPassword=<Password for CAPIF Privileged User>  --from-
literal=mysql-username=<CAPIF Privileged User Name> --from-literal=mysql-
password=<Password for CAPIF Privileged User> -n <Namespace of CAPIF
deployment>
```

> **Note**
>
> Note down the command used during the creation of kubernetes secret. This command is used for updating the secrets in future.

For example:

```
$ kubectl create secret generic privilegeduser-secret --from-
literal=dbUsername=occapifprivilegedusr --from-
literal=dbPassword=occapifprivilegedpasswd  --from-literal=mysql-
username=occapifprivilegedusr --from-literal=mysql-
password=occapifprivilegedpasswd -n occapif_namespace
```

2. Verify the secret creation with the following command:

```
$ kubectl describe secret <privileged user secret name> -n <Namespace of
occapif deployment>
```

For example:

```
$ kubectl describe secret privilegeduser-secret -n occapif_namespace
```

4.1.1.4.2 Creating and Updating Secret for Application Database User

This section explains how to create and update Kubernetes secret for application user to access the database.

Create Kubernetes secret for CAPIF application user with the following steps:

1. Run the following command to create Kubernetes secret:

```
$ kubectl create secret generic <appuser-secret name> --from-
literal=dbUsername=<CAPIF APPLICATION User Name> --from-
literal=dbPassword=<Password for CAPIF APPLICATION User>  --from-
literal=mysql-username=<CAPIF APPLICATION User Name> --from-literal=mysql-
password=<Password for CAPIF APPLICATION User> -n <Namespace of CAPIF
deployment>
```

> **Note**
>
> Note down the command used during the creation of Kubernetes secret. This command is used for updating the secrets in future.

For example:

```
$ kubectl create secret generic appuser-secret --from-
literal=dbUsername=occapifusr --from-literal=dbPassword=occapifpasswd --from-
literal=mysql-username=occapifusr --from-literal=mysql-password=occapifpasswd
-n occapif_namespace
```

2. Verify the secret creation with the following command:

```
$ kubectl describe secret <appuser-secret name> -n <Namespace of CAPIF
deployment>
```

For example:

```
$ kubectl describe secret appuser-secret -n occapif_namespace
```

4.1.1.4.3 Creating and Updating Secret for Storing Security Certificates for CAPIF

To create a Kubernetes secret for storing security certificates for CAPIF, perform the following steps:

1. Run the following command to create Kubernetes secret:

```
$ kubectl create secret generic <capif secret name> --from-
file=rsa_private_key_pkcs1.pem --from-file=trust.txt --fromfile=key.txt --
from-file=cert.cer --from-file=caroot.cer -n<namespace>
```

where,

`<secret name>` can be `ext-capif-secret` or `network-service-secret`, as required

rsa_private_key_pkcs1.pem is rsa private key

trust.txt is trust store password

key.txt is key store password

caroot.cer is the cer chain for trust store

cert.cer is signed server certificate

4.1.1.5 Configuring Secrets for Enabling HTTPS

This section describes the procedure to configure Kubernetes secrets for setting up secured communication of CAPIF with Ingress and Egress gateways.

> **Note**
>
> The passwords for TrustStore and KeyStore are stored in respective password files as mentioned in this section.

To create Kubernetes secret for HTTPS, following files are required:

- ECDSA private key and CA signed certificate of CAPIF, if initialAlgorithm is ES256
- RSA private key and CA signed certificate of CAPIF, if initialAlgorithm is RS256
- TrustStore password file
- KeyStore password file

> **Note**
>
> It is at the discretion of user to create the private keys and certificates, and it is not in the scope for NEF. This section lists only samples to create KeyPairs and certificates.

Update Secrets

This section explains how to update the secret with updated details.

1. Copy the exact command used in above section during creation of secret.
2. Update the same command with string "--dry-run -o yaml" and "kubectl replace -f - -n <Namespace of CAPIF deployment>".
   Example of the command syntax:

   ```
   $ kubectl create secret generic <secret-name> --from-
   file=<ssl_ecdsa_private_key.pem> --from-file=<rsa_private_key_pkcs1.pem> --
   from-file=<ssl_truststore.txt> --from-file=<ssl_keystore.txt> --from-
   file=<caroot.cer> --from-file=<ssl_rsa_certificate.crt> --from-
   file=<ssl_ecdsa_certificate.crt> --dry-run -o yaml -n <Namespace of capif
   deployment> | kubectl replace -f - -n <Namespace of capif deployment>
   ```

   Example:

   The names used below are same as provided in `oc-capif-24.2.2-custom-values.yaml` in CAPIF deployment:

   ```
   $ kubectl create secret generic ocegress-secret --from-
   file=ssl_ecdsa_private_key.pem --from-file=rsa_private_key_pkcs1.pem --
   from-file=ssl_truststore.txt --from-file=ssl_keystore.txt --from-
   file=caroot.cer --from-file=ssl_rsa_certificate.crt --from-
   file=ssl_ecdsa_certificate.crt --dry-run -o yaml -n occapif | kubectl
   replace -f - -n occapif
   ```

3. Run the updated command.
   After successful secret update, the following message is displayed:

   ```
   secret/<ocegress-secret> replaced
   ```

4.1.1.6 Configuring Secrets to Enable Access Token

This section explains how to configure a secret for enabling access token.

4.1.1.6.1 Generating Private Keys and Certificates

This section describes the procedure to generate private key and certificate.

Create Private Keys and Certificates

1. Generate RSA private key by running the following command:

   ```
   openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout
   rsa_private_key -out rsa_certificate.crt
   ```

2. Convert the private key to `.pem` format with the following command:

   ```
   openssl rsa -in rsa_private_key -outform PEM -out rsa_private_key_pkcs1.pem
   ```

3. Generate cert out of private key, by executing:

   ```
   openssl req -new -key rsa_private_key -out tmp.csr -config ssl.conf
   ```

> **Note**
>
> The `ssl.conf` can be used to configure default entries along with SAN details for your cert.

The following snippet shows a sample of the `ssl.conf` syntax:

```
#ssl.conf
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
req_extensions = req_ext

[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = IN
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Karnataka
localityName = Locality Name (eg, city)
localityName_default = Bangalore
organizationName = Organization Name (eg, company)
organizationName_default = Oracle
commonName = Common Name (e.g. server FQDN or YOUR name)
```

```
commonName_max = 64
commonName_default = localhost

[ req_ext ]
subjectAltName = @alt_names

[alt_names]
IP = 127.0.0.1
DNS.1 = localhost
```

4. Create Root CA by running the following command:

```
openssl req -new -keyout cakey.pem -out careq.pem
openssl x509 -signkey cakey.pem -req -days 3650 -in careq.pem -out
caroot.cer -extensions v3_ca
echo 1234 > serial.txt
```

5. Sign server cert with root CA private key by running the following command:

```
openssl x509 -CA caroot.cer -CAkey cakey.pem -CAserial serial.txt -req -in
tmp.csr -out tmp.cer -days 365 -extfile ssl.conf -extensions req_ext
```

> **Note**
>
> The `ssl.conf` file must be reused, as SAN contents is not packaged when signing.

4.1.1.7 Configuring Network Policies for CAPIF

Perform the following installation and upgrade procedures to deploy network policies for CAPIF.

> **Note**
>
> - Ports mentioned in policies are container ports and not the exposed service ports.
> - Connections that were created before installing network policy and still persist are not impacted by the new network policy. Only the new connections would be impacted.
> - If you are using ATS suite along with network policies, it is required to install the NEF, CAPIF, and ATS in the same namespace.
> - If the traffic is blocked or unblocked between the pods even after applying network policies, check if any existing policy is impacting the same pod or set of pods that might alter the overall cumulative behavior.
> - If changing default ports of services such as Prometheus, Database, Jaegar, or if Ingress or Egress Gateway names is overridden, update them in the corresponding network policies.
> - While using the debug container, it is recommended to uninstall the network policies or update them as required to establish the connections.

Configuring Network Policies

Following are the various operations that can be performed for network policies:

4.1.1.7.1 Installing Network Policy

Prerequisite

Network Policies are implemented by using the network plug-in. To use network policies, you must be using a networking solution that supports Network Policy.

> **Note**
>
> For a fresh installation, it is recommended to install Network Policies before installing NEF. However, if NEF is already installed, you can still install the Network Policies.

To install network policy

1. Open the `occapif-network-policy-custom-values-24.2.2.yaml` file provided in the release package zip file. For downloading the file, see Installation Package Download.
2. The custom values are provided with the default security policies.
   If required, update the `occapif-network-policy-custom-values-24.2.2.yaml` file as described in the Configurable Parameters of CAPIF section.
3. Run the following command to install the network policy:

   ```
   helm install <helm-release-name> occapif-network-policy/ -n <namespace> -f
   <custom-value-file>
   ```

   Sample command:
   ```
   helm install occapif-network-policy occapif-network-policy/ -n occapif -f
   occapif-network-policy-custom-values-24.2.2.yaml
   ```

   Where,

`helm-release-name`: occapif-network-policy Helm release name

`custom-value-file`: occapif-network-policy custom value file

`namespace`: namespace must be the OCCAPIF's namespace

4.1.1.7.2 Upgrading Network Policy

1. Modify the `occapif-network-policy-custom-values-24.2.2.yaml` file to add new network policies or update the existing policies.
2. Run the following command to upgrade the network policy:

   ```
   helm upgrade <helm-release-name> occapif-network-policy/ -n <namespace> -f
   <custom-value-file>
   ```

   Sample command:
   ```
   helm upgrade occapif-network-policy occapif-network-policy/ -n occapif -f
   occapif-network-policy-custom-values-24.2.2.yaml
   ```

   Where,
   `helm-release-name`: occapif-network-policy helm release name

   `custom-value-file`: occapif-network-policy custom value file

   `namespace`: namespace must be the CAPIF's namespace

4.1.1.7.3 Verifying Network Policies

Run the following command to verify if the network policies are deployed successfully:

```
kubectl get <helm-release-name> -n <namespace>
```

For example:

```
kubectl get occapif-network-policy -n occapif
```

where,

- `helm-release-name`: occapif-network-policy Helm release name.
- `namespace`: CAPIF namespace.

4.1.1.7.4 Uninstalling Network Policy

Run the following command to uninstall all network policies:

```
helm uninstall <helm-release-name> -n <namespace>
```

Sample command:

```
helm uninstall occapif-network-policy -n occapif
```

4.1.1.7.5 Configuration Parameters for Network Policies

Table 4-1 Configuration Parameters for Network Policy

Parameter	Description	Details

networkPolicies	The networkPolicies parameter is of array type. Each element of this array must have standard object's metadata and specification of the desired behavior for this NetworkPolicy.	This is an optional parameter. The network policy Helm chart creates policies for each entry.
		Example:
		`- metadata:`
		` name: <name of network policy>`
		`spec:`
		` <provide the standard specifications>`
		Note: Specify the policies compatible with apiversion networking.k8s.io/v1.

For more information about this functionality, see Network Policies in the *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

4.1.2 Installation Tasks

This section describes how to install CAPIF.

> **Note**
>
> Before installing CAPIF, you must complete Prerequisites and Preinstallation Tasks. In a georedundant deployment, perform the steps explained in this section on all georedundant sites.

4.1.2.1 Pushing the Images to Customer Docker Registry

CAPIF deployment package includes ready-to-use images and Helm charts to orchestrate containers in Kubernetes.

The following table lists the docker images for CAPIF:

Table 4-2 Docker Images for CAPIF

Service Name	Docker Image Name	Image Tag
CAPIF AF Manager	oc_capif_afmgr	24.2.2
CAPIF API Manager	oc_capif_apimgr	24.2.2
CAPIF Event Manager	oc_capif_eventmgr	24.2.2
Configuration Update Service	configurationupdate	24.2.15
Configuration INIT Service	configurationinit	24.2.15

Table 4-2 (Cont.) Docker Images for CAPIF

Service Name	Docker Image Name	Image Tag
Ingress Gateway	ocingress_gateway	24.2.15
Egress Gateway	ocegress_gateway	24.2.15
Debug Tools Service	ocdebug-tools	24.2.6
NF Test Service	nf_test	24.2.5
Console Data Service	oc_nef_console_data_service	24.2.2

Pushing Docker Images

Prerequisite: Download and untar the NEF Package ZIP file. For more information about downloading the package, see Installation Package Download.

To push the images to the registry:

1. Unzip the release package to the location where you want to install NEF. The package is as follows:
   `ocnef-pkg-24.2.2.0.0.tgz`
2. Untar the NEF package zip file to get NEF image tar file:

   ```
   tar -xvzf ocnef-pkg-24.2.2.0.0.tgz
   ```

   The directory consists of the following:
   - `occapif-24.2.2.tgz`: CAPIF Helm Chart
   - `occapif-24.2.2.tgz.sha256`: Checksum for Helm chart tgz file
   - `occapif-images-24.2.2.tar`: CAPIF Images File
   - `occapif-images-24.2.2.tar.sha256`: Checksum for images tar file
   - `occapif-network-policy-24.2.2.tgz`: CAPIF Helm Chart for network policy
   - `occapif-network-policy-24.2.2.tgz.sha256` Checksum for network policy Helm chart tgz file
3. Run one of the following commands to load the `occapif-images-24.2.2.tar` file:

   ```
   docker load --input /IMAGE_PATH/occapif-images-24.2.2.tar
   ```

   ```
   podman load --input /IMAGE_PATH/occapif-images-24.2.2.tar
   ```
4. Run one of the following commands to verify the images are loaded:

   ```
   docker images
   ```

   ```
   podman images
   ```

> **Note**
>
> Verify the list of images shown in the output with the list of images shown in the above table. If the list does not match, reload the image tar files.

5. Run one of the following commands to tag the images to the registry:

```
docker tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
```

```
podman tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
```

6. Run one of the following commands to push the images to the registry:

```
docker push <docker-repo>/<image-name>:<image-tag>
```

```
podman push <docker-repo>/<image-name>:<image-tag>
```

> **Note**
>
> It is recommended to configure the Docker certificate before running the push command to access customer registry through HTTPS, otherwise, docker push command may fail.

4.1.2.2 Pushing the CAPIF Images to OCI Registry

CAPIF deployment package includes ready-to-use images and Helm charts to orchestrate containers in Kubernetes.

The following table lists the docker images for CAPIF:

Table 4-3 Docker Images for CAPIF

Service Name	Docker Image Name	Image Tag
CAPIF AF Manager	oc_capif_afmgr	24.2.2
CAPIF API Manager	oc_capif_apimgr	24.2.2
CAPIF Event Manager	oc_capif_eventmgr	24.2.2
Configuration Update Service	configurationupdate	24.2.15
Configuration INIT Service	configurationinit	24.2.15
Ingress Gateway	ocingress_gateway	24.2.15
Egress Gateway	ocegress_gateway	24.2.15
Debug Tools Service	ocdebug-tools	24.2.6
NF Test Service	nf_test	24.2.5

Pushing Docker Images

Prerequisite: Download and untar the NEF Package ZIP file. For more information about downloading the package, see Installation Package Download.

To push the images to the registry:

1. Unzip the release package to the location where you want to install NEF. The package is as follows:
   `ocnef-pkg-24.2.2.0.0.tgz`
2. Untar the NEF package zip file to get NEF image tar file:

   ```
   tar -xvzf ocnef-pkg-24.2.2.0.0.tgz
   ```

   The directory consists of the following:
   - `occapif-24.2.2.tgz`: CAPIF Helm Chart
   - `occapif-24.2.2.tgz.sha256`: Checksum for Helm chart tgz file
   - `occapif-images-24.2.2.tar`: CAPIF Images File
   - `occapif-images-24.2.2.tar.sha256`: Checksum for images tar file
   - `occapif-network-policy-24.2.2.tgz`: CAPIF Helm Chart for network policy
   - `occapif-network-policy-24.2.2.tgz.sha256` Checksum for network policy Helm chart tgz file
3. Run one of the following commands to load the `occapif-images-24.2.2.tar` file:

   ```
   docker load --input /IMAGE_PATH/occapif-images-24.2.2.tar
   ```

   ```
   podman load --input /IMAGE_PATH/occapif-images-24.2.2.tar
   ```
4. Run one of the following commands to verify the images are loaded:

   ```
   docker images
   ```

   ```
   podman images
   ```

   > **Note**
   >
   > Verify the list of images shown in the output with the list of images shown in the above table. If the list does not match, reload the image tar files.
5. Run one of the following commands to tag the images to the registry:

   ```
   docker tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
   ```

   ```
   podman tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
   ```
6. Run one of the following commands to push the images to the registry:

   ```
   docker push <docker-repo>/<image-name>:<image-tag>
   ```

   ```
   podman push <docker-repo>/<image-name>:<image-tag>
   ```

> **Note**
>
> It is recommended to configure the Docker certificate before running the push command to access customer registry through HTTPS, otherwise, docker push command may fail.

4.1.2.3 Installing CAPIF Package

To install the CAPIF package, perform the following steps:

1. Run the following command to access the extracted package:

   ```
   cd occapif-<release_number>
   ```

   For example:

   ```
   cd occapif-24.2.2.0.0
   ```

2. Customize the oc-capif-24.2.2-custom-values.yaml file with the required deployment parameters. See Customizing CAPIF chapter to customize the file. For more information about predeployment parameter configurations, see Preinstallation.
3. Run the following command to install CAPIF:

   ```
   helm install -f <custom-file> <release_name> <helm-chart> --namespace
   <release_namespace> --timeout 10m
   ```

   For example:

   ```
   helm install -f occapif-24.2.2-custom-values-occapif.yaml occapif /home/
   cloud-user/occapif-24.2.2.tgz --namespace occapif
   ```

   where:

   - *helm_chart* is the location where helm charts are stored.
   - occapif-24.2.2.tgz is the helm chart.
   - *release_name* is the release name used by helm command.

     > **Note**
     >
     > – The *release_name* should not exceed 63 characters.
     >
     > – In case of georedundant setup, it is mandatory to use unique *release_name* for each CAPIF instance.

   - *release_namespace* is the deployment namespace used by helm command.
   - *custom-file* is the name of the custom values yaml file (including location).

> **Note**
>
> - You can verify the installation while running the install command by entering this command on a separate terminal:
>
>   ```
>   watch kubectl get jobs,pods -n release_namespace
>   ```
>
> - The DB hooks start creating the CAPIF database tables, once the `helm install` command is run.

Following are the optional parameters that can be used in the `helm install` command:

- **atomic:** If this parameter is set, installation process purges chart on failure. The `--wait` flag will be set automatically.
- **wait:** If this parameter is set, installation process will wait until all pods, PVCs, Services, and minimum number of pods of a deployment, StatefulSet, or ReplicaSet are in a ready state before marking the release as successful. It will wait for as long as `--timeout`.
- **timeout duration:** If not specified, default value will be 300 seconds in Helm. It specifies the time to wait for any individual Kubernetes operation (like Jobs for hooks). If the `helm install` command fails at any point to create a Kubernetes object, it will internally call the purge to delete after the timeout value. Here, the timeout value is not for overall installation, but for automatic purge on installation failure.

> **Caution**
>
> Do not exit from `helm install` command manually. After running the `helm install` command, it takes some time to install all the services. In the meantime, you must not press **Ctrl+C** to come out from the command. It may lead to some anomalous behavior.

4. Press **Ctrl+C** to exit watch mode. Run the `watch` command on another terminal. Run the following command to check the status:

   ```
   helm status release_name -n release_namespace
   ```

4.1.3 Postinstallation Tasks

This section explains the postinstallation tasks for CAPIF.

4.1.3.1 Verifying CAPIF Installation

To verify the CAPIF installation:

1. Run the following command:

   ```
   helm status <helm-release> -n <namespace>
   ```

   Where,

   `<helm-release>` is the Helm release name of CAPIF.

<namespace> is the namespace of CAPIF deployment.

For example:

```
helm status occapif -n occapif_namespace
```

In the output, if STATUS is showing as deployed, then the installation is successful.

Sample output:

```
NAME: occapif
LAST DEPLOYED: Fri Sep 18 10:08:03 2020
NAMESPACE: occapif
STATUS: deployed
 REVISION: 1
```

2. Run the following command to verify if the pods are up and active:

```
kubectl get jobs,pods -n <Namespace>
```

Where,

<Namespace> is the namespace where CAPIF is deployed.

For example:

```
kubectl get pod -n occapif
```

In the output, the STATUS column of all the pods must be Running and the READY column of all the pods must be n/n, where n is the number of containers in the pod.

3. Run the following command to verify if the services are deployed and active:

```
kubectl get services -n <Namespace>
```

For example:

```
kubectl get services -n occapif_namespace
```

> **Note**
>
> If the installation is unsuccessful or the STATUS of all the pods is not in the Running state, perform the troubleshooting steps provided in the *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

4.1.3.2 Performing Helm Test

This section describes how to perform sanity check for CAPIF installation through Helm test. The pods to be checked should be based on the namespace and label selector configured for the Helm test configurations.

Helm Test is a feature that validates successful installation of CAPIF and determines if the NF is ready to take traffic.

> **Note**
>
> Helm Test expects all of the pods of given microservice to be in `READY` state for a successful result. However, the NRF Client Management microservice comes with Active/Standby mode for the multi-pod support in the current release. When the multi-pod support for NRF Client Management service is enabled, you may ignore if the Helm Test for NRF-Client-Management pod fails.

1. Complete the Helm test configurations under the "Helm Test Global Parameters" section of the `oc-capif-24.2.2-custom-values.yaml` file.

```
nfName: occapif
    image:
      name: nf_test
      tag: 24.2.2
      registry: cgbu-cnc-comsvc-release-docker.dockerhub-
phx.oci.oraclecorp.com/cgbu-ocudr-nftest
    config:
      logLevel: WARN
      timeout: 120      #Beyond this duration helm test will be considered
failure
    resources:
    - horizontalpodautoscalers/v1
    - deployments/v1
    - configmaps/v1
    - prometheusrules/v1
    - serviceaccounts/v1
    - poddisruptionbudgets/v1
    - roles/v1
    - statefulsets/v1
    - persistentvolumeclaims/v1
    - services/v1
    - rolebindings/v1
    complianceEnable: true
```

   For more information on Helm test parameters, see Global Parameters.

2. Run the following command to perform the helm test:

```
helm test <release_name> -n <namespace>
```

   where:

   <release_name> is the release name.

   namespace is the deployment namespace where CAPIF is installed.

   For example:

```
helm test occapif -n occapif
```

Sample output:

```
NAME: occapif
LAST DEPLOYED: Fri Nov 12 10:08:03 2020
NAMESPACE: occapif
STATUS: deployed
REVISION: 1
TEST SUITE:     occapif-test
Last Started:   Fri Nov 12 10:41:25 2020
Last Completed: Fri Nov 12 10:41:34 2020
Phase:          Succeeded
```

If the Helm test fails, see *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

4.1.3.3 Taking a Backup

Take a backup of the following files, which are required during fault recovery:

- Updated oc-capif-24.2.2-custom-values.yaml file
- Updated helm charts
- Secrets, certificates, and keys that are used during installation

4.2 Customizing CAPIF

This section provides information about customizing CAPIF deployment in a cloud native environment.

The CAPIF deployment is customized by overriding the default values of various configurable parameters in the oc-capif-24.2.2-custom-values.yaml file.

Basic Configurations

- Once Docker platform configurations are done, proceed as per Configurable Parameters of CAPIF.
- Check Registry is in place and contains latest helm charts and jar as per the release for CAPIF node.

Perform the following steps to customize the oc-capif-24.2.2-custom-values.yaml file as per the required parameters: :

1. Unzip the `Custom_Templates` file available in the extracted documentation release package to get the following file that is used to customize the deployment parameters during installation:
   - oc-capif-24.2.2-custom-values.yaml: This file is used to customize the deployment parameters during installation.
   - oc-capif-24.2.2-custom-values-ats.yaml: This file is used to customize the CAPIF deployment with ATS.
   - CapifAlertrules-24.2.2.yaml: This file is used for `Prometheus`.
   - `ocnef_dbtier_24.2.2_custom_values_24.2.2.yaml`: This file is used to install NEF with the recommended resource requirements for cnDBTier.

   For more information on how to download the package from My Oracle Support, see Installation Package Download section.

2. Customize the oc-capif-24.2.2-custom-values.yaml file.
3. Save the updated oc-capif-24.2.2-custom-values.yaml file in the helm chart directory.

> **Note**
>
> - All parameters mentioned as mandatory must be present in Custom Values YAML file and configured before the deployment.
> - All fixed value parameters listed must be present in the Custom Values YAML file with the exact values as specified in this section.

For more information on the configurable parameters, see Configurable Parameters of CAPIF.

4.2.1 Configurable Parameters of CAPIF

This section includes information about the configuration parameters of CAPIF.

CAPIF allows customization of parameters for the different microservices and related settings.

> **Note**
>
> - Mandatory parameters must be configured before the CAPIF deployment.
> - During installation, all the configurations would be read from Helm. Any configurations that support **Update** operation through REST API can only be updated using REST API or Console and any further updates to these configurations using Helm would be ignored. For further information, refer to *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide* and *Configuring Network Exposure Function using the CNC Console* chapter in *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

4.2.1.1 Global Parameters

Table 4-4 Global Paramaters

Parameter	Description	Details
`dockerRegistry`	Specifies the name of the Docker registry, which hosts CNC Policy docker images.	This is a docker registry running CNE bastion server where all OAuth docker images are loaded. This is a mandatory parameter.
`vendor`	The vendor name.	This is a mandatory parameter. **Default Value:** Oracle
`serviceAccountName`	Name of the service account for CAPIF.	This is an optional parameter.
`app_name`	Name of the application.	This is a mandatory parameter. **Default Value:** occapif

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
`capifK8sNameSpace`	The Kubernetes namespace.	**Default Value:** &capifNameSpace occapif
`capifInstanceId`	The unique instance identifier of CAPIF.	This is a mandatory parameter. The parameter must have unique for each CAPIF instance in a georedundant deployment.
`capifApiPrefix`	This is the prefix set for all CAPIF APIs.	This is a mandatory parameter. **Default Value:** ' '
Jaeger Tracing Configurations		
`jaegerTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing	**Default Value:** false
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5
`openTelemetry.jaeger.httpExporter.host`	Specifies the host of Jaeger collector service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger collector service	**Default Value:** 4318
mTLS Configurations - External Gateways		
`externalGWConfig.initSSL`	Enable this to enable mtls on external gateways for CAPIF that includes both Ingress GW (incoming) and Egress GW (outgoing) of CAPIF.	This value must be set to **true** only if the value for the `externalGWConfig.igw.enableIncomingHttp` parameter is **false**. **Default Value:** true This is a mandatory parameter. **Note:** It is mandatory to configure all the parameters under `externalGWConfig` for CAPIF, irrespective of the value of `externalGWConfig.initSSL`.
`externalGWConfig.igw.enableIncomingHttp`	Enables HTTP requests on northbound side (extEnableIncomingHttp)	This value can be set to **true** only if the value for the `externalGWConfig.initSSL` parameter is **false**. **Default Value:** false This is a mandatory parameter. **Note:** Either `externalGWConfig.initSSL` or `externalGWConfig.igw.enableIncomingHttp` must be set to **true**.

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
`externalGWConfig.igw.publicHttpSignallingPort`		**Default Value:** 80 This is a mandatory parameter.
`externalGWConfig.igw.publicHttpsSignallingPort`		**Default Value:** 443 This is a mandatory parameter.
`externalGWConfig.egw.publicHttpsSignallingPort`		**Default Value:** 8080 This is a mandatory parameter.
`externalGWConfig.tls.privateKey.k8SecretName`	Name of the Kubernetes secret object containing ext-capif-secret username and password	This is an optional parameter. **Default Value:** ext-capif-secret
`externalGWConfig.tls.privateKey.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-capif-secret.	This is an optional parameter.
`externalGWConfig.tls.privateKey.rsa.filename`	The filename containing rsa keydetail of the ext-capif-secret.	This is an optional parameter. **Default Value:** rsa_private_key_pkcs1.pem
`externalGWConfig.tls.privateKey.ecdsa.filename`	The filename containing ecdsa keydetail of the ext-capif-secret.	This is an optional parameter. **Default Value:** ssl_ecdsa_private_key.pem
`externalGWConfig.tls.certificate.k8SecretName`	Name of the Kubernetes secret object containing ext-capif-secret certificate.	This is an optional parameter. **Default Value:** ext-capif-secret
`externalGWConfig.tls.certificate.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-capif-secret certificate.	This is a mandatory parameter.
`externalGWConfig.tls.certificate.rsa.filename`	The filename containing rsa keydetail of the ext-capif-secret certificate.	This is a mandatory parameter. **Default Value:** tmp.cer
`externalGWConfig.tls.certificate.ecdsa.filename`	The filename containing ecdsa keydetail of the ext-capif-secret certificate.	This is a mandatory parameter. **Default Value:** ssl_ecdsa_certificate.crt
`externalGWConfig.tls.caBundle.k8SecretName`	Name of the Kubernetes secret object containing ext-capif CA details for truststore.	This is a mandatory parameter. **Default Value:** ext-capif-secret
`externalGWConfig.tls.caBundle.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-capif CA details for truststore.	This is a mandatory parameter.
`externalGWConfig.tls.caBundle.filename`	The filename containing ext-capif CA details for truststore.	This is a mandatory parameter. **Default Value:** caroot.cer
`externalGWConfig.tls.keyStorePassword.k8SecretName`	Name of the Kubernetes secret object containing ext-capif KeyStore password	This is a mandatory parameter. **Default Value:** ext-capif-secret

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
`externalGWConfig.tls.keyStorePassword.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-capif KeyStore password	This is a mandatory parameter.
`externalGWConfig.tls.keyStorePassword.filename`	The filename containing ext-capif KeyStore password	This is a mandatory parameter. **Default Value:** key.txt
`externalGWConfig.tls.trustStorePassword.k8SecretName`	Name of the Kubernetes secret object containing ext-capif truststore password	This is a mandatory parameter. **Default Value:** ext-capif-secret
`externalGWConfig.tls.trustStorePassword.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-capif truststore password	This is a mandatory parameter.
`externalGWConfig.tls.trustStorePassword.filename`	The filename containing ext-capif truststore password	This is a mandatory parameter. **Default Value:** trust.txt
`externalGWConfig.tls.initialAlgorithm`	The initial algorithm selected by ext-capif.	Possible values are: • ES256 • RS256 **Default Value:** RS256 This is a mandatory parameter.
`externalGWConfig.tls.keyType`	The selected key type.	Possible values are: • rsakey • ecdsakey **Default Value:** rsakey This is a mandatory parameter.
mTLS Configurations - Network Gateways		
`networkGWConfig.initSSL`	Enable this to enable mTLS on 5GC gateways that includes both Ingress GW (incoming) and Egress GW (outgoing) of CAPIF.	This value must be set to **true** only if the value for the `networkGWConfig.igw.enableIncomingHttp` parameter is **false**. **Default Value:** true This is a mandatory parameter.

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
`networkGWConfig.igw.enableIncomingHttp`	Enables HTTP requests on northbound side (extEnableIncomingHttp)	This value can be set to **true** only if the value for the `networkGWConfig.initSSL` parameter is **false**. **Default Value:** false This is a mandatory parameter. **Note:** Either `networkGWConfig.initSSL` or `networkGWConfig.igw.enableIncomingHttp` must be set to **true**.
`networkGWConfig.igw.publicHttpSignallingPort`		**Default Value:** 80 This is a mandatory parameter.
`networkGWConfig.igw.publicHttpsSignallingPort`		**Default Value:** 443 This is a mandatory parameter.
`networkGWConfig.egw.publicHttpsSignallingPort`		**Default Value:** 8080 This is a mandatory parameter.
`networkGWConfig.tls.privateKey.k8SecretName`	Name of the Kubernetes secret object containing network-service-secret username and password	This is an optional parameter. **Default Value:** network-service-secret This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**.
`networkGWConfig.tls.privateKey.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-network-service-secret.	This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**.
`networkGWConfig.tls.privateKey.rsa.filename`	The filename containing rsa keydetail of the network-service-secret.	This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**. **Default Value:** rsa_private_key_pkcs1.pem
`networkGWConfig.tls.privateKey.ecdsa.filename`	The filename containing ecdsa keydetail of the capif-secret.	This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**. **Default Value:** ssl_ecdsa_private_key.pem

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
networkGWConfig.tls.certificate.k8SecretName	Name of the Kubernetes secret object containing capif-secret certificate.	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**. **Default Value:** network-service-secret
networkGWConfig.tls.certificate.k8NameSpace	The namespace of the Kubernetes secret object containing network-service-secret certificate.	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**.
networkGWConfig.tls.certificate.rsa.filename	The filename containing rsa keydetail of the network-service-secret certificate.	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**. **Default Value:** tmp.cer
networkGWConfig.tls.certificate.ecdsa.filename	The filename containing ecdsa keydetail of the network-service-secret certificate.	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**. **Default Value:** ssl_ecdsa_certificate.crt
networkGWConfig.tls.caBundle.k8SecretName	Name of the Kubernetes secret object containing network-service-secret CA details for truststore.	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**. **Default Value:** network-service-secret
networkGWConfig.tls.caBundle.k8NameSpace	The namespace of the Kubernetes secret object containing network-service-secret CA details for truststore.	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**.
networkGWConfig.tls.caBundle.filename	The filename containing network-service-secret CA details for truststore.	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**. **Default Value:** caroot.cer
networkGWConfig.tls.keyStorePassword.k8SecretName	Name of the Kubernetes secret object containing network-service-secret KeyStore password	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**. **Default Value:** network-service-secret
networkGWConfig.tls.keyStorePassword.k8NameSpace	The namespace of the Kubernetes secret object containing network-service-secret KeyStore password	This is a mandatory parameter when the value for networkGWConfig.initSSL is set to **true**.

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
`networkGWConfig.tls.keyStorePassword.filename`	The filename containing network-service-secret KeyStore password	This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**. **Default Value:** network-service-secret
`networkGWConfig.tls.trustStorePassword.k8SecretName`	Name of the Kubernetes secret object containing network-service-secret truststore password	This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**.
`networkGWConfig.tls.trustStorePassword.k8NameSpace`	The namespace of the Kubernetes secret object containing network-service-secret truststore password	This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**. **Default Value:**&trustStorePasswdSecretNameSpace default
`networkGWConfig.tls.trustStorePassword.filename`	The filename containing network-service-secret truststore password	This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**. **Default Value:**&trustStorePasswdFileName trust.txt
`networkGWConfig.tls.initialAlgorithm`	The initial algorithm selected by network-service-secret.	Possible values are: • ES256 • RS256 **Default Value:** RS256 This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**.
`networkGWConfig.tls.keyType`	The selected key type.	Possible values are: • rsakey • ecdsakey **Default Value:** rsakey This is a mandatory parameter when the value for `networkGWConfig.initSSL` is set to **true**.
Database Configurations		
`database.dbName`	Name of the CAPIF service database	This is a mandatory parameter. **Note:** The parameter value must be different for all the CAPIF instances in a georedundant depolyment.

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
`database.releaseDbName`	Name of the release database containing release version details	This is a mandatory parameter. **Default Value:** capifcore **Note:** The parameter value must be different for all the CAPIF instances in a georedundant depolyment.
`database.dbPrimaryHost`	The primary host details for the database	This is a mandatory parameter.
`database.dbSecondaryHost`	The secondary host details for the database	This is a mandatory parameter.
`database.dbPort`	Database Port details	
`database.appUserSecretName`	Name of the Kubernetes secret object containing Database username and password for an application user	This is a mandatory parameter. **Default Value:** appuser-secret
`database.privilegedUserSecretName`	Name of the Kubernetes secret object containing Database username and password for an privileged user	This is a mandatory parameter. **Default Value:** privilege-user-secret
`database.dbUNameLiteral`	Name of the key configured for "DB Username" in appuser-secret.	This is a mandatory parameter.
`database.dbPwdLiteral`	Name of the key configured for "DB Password" in appuser-secret.	This is a mandatory parameter.
`database.engine`	The database engine.	Any value other than InnoDB or NDBCluster leads to failure of database table creation process. This is a mandatory parameter. **Default Value:** NDBCluster
`extraContainers`	The flag to enable or disable injecting extra container.	This is an optional parameter. **Default Value:** DISABLED
`debugToolContainerMemoryLimit`	Indicates the memory assigned for the debug tool container.	
`ingressCommonSvcName`		**Default Value:** igw
`egressCommonSvcName`		**Default Value:** egw
Prometheus Scraping Configuration		
`prometheusScrapingConfig.enabled`	Flag to enable or disable the Prometheus Scraping Configuration.	This is a mandatory parameter. **Default Value:** true
`prometheusScrapingConfig.path`	Prometheus scrap path	This is a mandatory parameter. **Default Value:** "/actuator/prometheus"

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
`type`	The type of service.	Possible values are: • ClusterIP • NodePort • LoadBalancer • ExternalName This is an optional parameter. **Default Value:** LoadBalancer
Custom Extension Global Configuration		
`customExtension.allResources.labels`	Custom Labels that needs to be added to all the CAPIF Kubernetes resources	This can be used to add custom label(s) to all Kubernetes resources that are created by CAPIF helm chart. This is an optional parameter.
`customExtension.allResources.annotations`	Custom Annotations that needs to be added to all the CAPIF Kubernetes resources	This can be used to add custom annotation(s) to all Kubernetes resources that are created by CAPIF helm chart. This is an optional parameter.
`customExtension.lbServices.labels`	Custom Labels that needs to be added to CAPIF services that are considered as Load Balancer type	This can be used to add custom label(s) to all Load Balancer Type Services that are created by CAPIF helm chart. This is an optional parameter.
`customExtension.lbServices.annotations`	Custom Annotations that needs to be added to CAPIF services that are considered as Load Balancer type	This can be used to add custom annotation(s) to all Load Balancer Type Services that are created by CAPIF helm chart. This is an optional parameter.
`customExtension.lbDeployments.labels`	Custom Labels that needs to be added to CAPIF deployments that are associated to a service which is of Load Balancer type	This can be used to add the custom label(s) to the deployments that will be created by CAPIF helm chart that are associated to a Load Balancer Type Service. This is an optional parameter.
`customExtension.lbDeployments.annotations`	Custom Annotations that needs to be added to CAPIF deployments that are associated to a service which is of Load Balancer type	This can be used to add the custom label(s) to the deployments that will be created by CAPIF helm chart that are associated to a Load Balancer Type Service. This is an optional parameter.

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details
customExtension.nonlbServices.labels	Custom Labels that needs to be added to CAPIF Services that are considered as not Load Balancer type	This can be used to add custom label(s) to all non-Load Balancer Type Services that is created by CAPIF helm chart. This is an optional parameter.
customExtension.nonlbServices.annotations	Custom Annotations that needs to be added to CAPIF Services that are considered as not Load Balancer type	This can be used to add custom annotation(s) to all non-Load Balancer Type Services that is created by CAPIF helm chart. This is an optional parameter.
customExtension.nonlbDeployments.labels	Custom Labels that needs to be added to CAPIF Deployments that are associated to a Service which is not of Load Balancer type	This can be used to add custom label(s) to all Deployments that is created by CAPIF helm chart which are associated to a Service which if not of Load Balancer Type. This is an optional parameter.
k8sResource.container.prefix	Value that is prefixed to all the container names of CAPIF.	
k8sResource.container.prefix	Value that is suffixed to all the container names of CAPIF.	
Helm Test Hook Configurations		
test.nfName	Name of the NF	This is a mandatory parameter. **Default Value:** occapif
test.image.name	Image name for the test container	This is a mandatory parameter. **Default Value:** nf_test
test.image.tag	Tag for the test container image	This is a mandatory parameter.
test.config.logLevel	Set the logging level	This is a mandatory parameter. **Default Value:** WARN
test.config.timeout	Specify timeout until test container checks for Pod's health	This is a mandatory parameter. **Default Value:** 20

Table 4-4 (Cont.) Global Paramaters

Parameter	Description	Details

`test.resources`	Specifies the helm test resources.	**Example:**
		`horizontalpodautoscalers/v1`
		`deployments/v1`
		`configmaps/v1`
		`prometheusrules/v1`
		`serviceaccounts/v1`
		`poddisruptionbudgets/v1`
		`roles/v1`
		`statefulsets/v1`
		`persistentvolumeclaims/v1`
		`services/v1`
		`rolebindings/v1`

`test.complianceEnable`	Enables or disables the helm test logging feature.	This is a mandatory parameter. Possible values are: • true • false **Default value:** false
Configurable Error Codes Configuration		
`configurableErrorCodes.enabled`	Specifies if the configurable error codes must be enabled.	This is a mandatory parameter. Possible values are: • true • false **Default value:** false

`configurableErrorCodes.errorScenarios`	Contains a list of exceptionType, error codes, errorDescription, error causes, and errorTitle for which a failover must be performed.	**Example:**
		`configurableErrorCodes:`
		`enabled: false`
		`errorScenarios:`
		`exceptionType:`
		`"ConnectException"`
		`errorCode: "503"`
		`errorDescription:`
		`"Connection failure"`
		`errorCause: "Connection Refused"`
		`errorTitle:`
		`"ConnectException"`

4.2.1.2 API Manager Parameters

Table 4-5 API Manager Parameters

Parameter	Description	Details
image.name	occapif-apimgr image name	This is an optional parameter. **Default Value:** oc_occapif_apimgr
image.tag	Tag name of image	This is an optional parameter. **Default Value:** CAPIF Images
image.pullPolicy	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
resources.limits.cpu	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
resources.limits.memory	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi
resources.requests.cpu	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
resources.requests.memory	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
resources.target.averageCpuUtil	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 60
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 3
maxUnavailable		**Default Value:** 25%
maxSurge		**Default Value:** 25%
readinessProbe.initialDelaySeconds	Informs the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
readinessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 3
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1

Table 4-5 (Cont.) API Manager Parameters

Parameter	Description	Details
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 3
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 5
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 5
`service.type`	Type of the service.	This is an optional parameter. **Default Value:** ClusterIP
`deployment.customExtension.labels`	Custom lables that needs to be added to API Manager service.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to API Manager service.	
`database.name`	The database name for the occapif-apimgr service	
`log.level.root`	Log level for root level logs	This is an optional parameter. **Default Value:** WARN
`log.level.apimgr`	Log level for apimgr level logs	This is an optional parameter. **Default Value:** INFO
`extraContainers`	Specifies if extra container must be used for DEBUG tool.	Possible Values are: • USE_GLOBAL_VALUE • DISABLED • ENABLED This is an optional parameter. **Default Value:** USE_GLOBAL_VALUE

4.2.1.3 AF Manager Parameters

Table 4-6 AF Manager Parameters

Parameter	Description	Details
`image.name`	Name of image.	This is an optional parameter. **Default Value:** oc_capif_afmgr
`image.tag`	Tag name of image.	This is an optional parameter. **Default Value:** CAPIF Images

Table 4-6 (Cont.) AF Manager Parameters

Parameter	Description	Details
`image.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`initContainersImage.name`	Image Name for AF Manager init container	This is an optional parameter. **Default Value:** configurationinit
`initContainersImage.tag`	Tag Name for AF Manager init container	This is an optional parameter. **Default Value:** CAPIF Images
`initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`updateContainersImage.name`	Image Name for update container	This is an optional parameter. **Default Value:** configurationupdate
`updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** CAPIF Images
`updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 60

Table 4-6 (Cont.) AF Manager Parameters

Parameter	Description	Details
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 3
maxUnavailable		**Default Value:** 25%
maxSurge		**Default Value:** 25%
readinessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
readinessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 3
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 3
livenessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
livenessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 5
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 5
deployment.customExtension.labels	Custom lables that needs to be added to AF Manager deployment.	
deployment.customExtension.annotations	Custom annotations that needs to be added to AF Manager deployment.	
initssl	Type of the service.	This value must always be set as true. **Default Value:** true
service.type	The type of the service.	
service.customExtension.labels	Custom lables that needs to be added to AF Manager service.	
service.customExtension.annotations	Custom annotations that needs to be added to AF Manager service.	
ssl.keyType	The selected key type.	
ssl.tlsVersion	The TLS version.	

Table 4-6 (Cont.) AF Manager Parameters

Parameter	Description	Details
`ssl.privateKey.k8SecretName`	Secret name that contains CAPIF AF Manager Private Key	
`ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	
`ssl.privateKey.rsa.filename`	CAPIF's Private Key (RSA type) file name	
`ssl.privateKey.ecdsa.filename`	CAPIF's Private Key (ECDSA type) file name	
`certificate.k8SecretName`	Secret name that contains CAPIF's certificate for HTTPS	
`certificate.k8NameSpace`	Namespace in which CAPIF's certificate is present	
`certificate.rsa.filename`	CAPIF's Certificate (RSA type) file name	
`certificate.ecdsa.filename`	CAPIF's Certificate (ECDSA type) file name	
`caBundle.k8SecretName`	Secret name that contains CAPIF's CA details for HTTPS	
`caBundle.k8NameSpace`	Namespace in which CAPIF's CA details is present	
`caBundle.filename`	CAPIF's CA bundle filename	
`keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	
`keyStorePassword.k8NameSpace`	Namespace in which CAPIF's keystore password is present	
`keyStorePassword.filename`	CAPIF's Key Store password Filename	
`trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`trustStorePassword.filename`	CAPIF's trustStorePassword Filename	
`ssl.initialAlgorithm`	Initial Algorithm for HTTPS	
`database.name`	The CAPIF AF Manager database name	
`extraContainers`	Specifies if extra container must be used for DEBUG tool.	Possible Values are: • USE_GLOBAL_VALUE • DISABLED • ENABLED This is an optional parameter. **Default Value:** USE_GLOBAL_VALUE
`log.level.root`	Log level for root logs	
`log.level.afmgr`	Log level for AF Manager service logs	
`log.level.updateContainer`	Log level for updateContainer logs	

Table 4-6 (Cont.) AF Manager Parameters

Parameter	Description	Details
accessToken.expiryTime	To set the validity of the token to access NEF services.	This is a mandatory parameter. **Default value:** 3600 seconds **Note:** • Maximum value can be 86400 sec. Even if a value greater than that is set, it is internally still considered as 86400 sec only. • While obtaining the access token for a API Invoker, if time (in sec) left in its expiry becomes lesser than configured expiryTime, then the system generates the token whose expiry is set to the time (in sec) that the API Invoker has left.

4.2.1.4 Event Manager Parameters

Table 4-7 Event Manager Parameters

Parameter	Description	Details
image.name	Name of image.	This is an optional parameter. **Default Value:** oc_capif_eventmgr
image.tag	Tag name of image.	This is an optional parameter. **Default Value:** CAPIF Images
image.pullPolicy	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
httpTwoEnabled		**Default Value:** true
log.level.root	Log level for root logs	**Default Value:** DEBUG
log.level.events	Log level for Event Manager service logs	**Default Value:** DEBUG
resources.limits.cpu	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
resources.limits.memory	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi

Table 4-7 (Cont.) Event Manager Parameters

Parameter	Description	Details
resources.requests.cpu	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
resources.requests.memory	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
resources.target.averageCpuUtil	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 60
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
maxUnavailable		**Default Value:** 25%
maxSurge		**Default Value:** 25%
readinessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 40
readinessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 3
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 3
livenessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 40
livenessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 5
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 5
extraContainers	Specifies if extra container must be used for DEBUG tool.	
deployment.customExtension.labels	Custom lables that needs to be added to Event Manager deployment.	

Table 4-7 (Cont.) Event Manager Parameters

Parameter	Description	Details
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Event Manager deployment.	
`service.type`	The type of the service.	
`service.customExtension.labels`	Custom labels that needs to be added to Event Manager service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Event Manager service.	
`auditconf.purgeExpSubsCron`	Set the Cron job scheduler to delete the expired events.	**Default Value:** "0 0 1 * * ?" (1 AM everyday)

4.2.1.5 Ingress Gateway Parameters

The following table describes the parameters for the External Ingress GW and Network Ingress GW services.

- External Ingress GW
- Network Ingress GW

External Ingress GW

Table 4-8 External Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`global.type`	The service type that will be used for this deployment.	It is not recommended to change the service type. **Default Value:** LoadBalancer
`global.staticIpAddressEnabled`	Specifies if static load balancer IP needs to be set	**Default Value:** false
`global.staticIpAddress`	Static IP address assigned to the Load Balancer from the external load balancer IP pool.	
`global.staticNodePortEnabled`	Specifies if static node port needs to be set	**Default Value:** false
`global.staticHttpNodePort`	Static HTTP Node Port	
`global.staticHttpsNodePort`	Static HTTPS Node Port	
`global.publicHttpSignalingPort`	HTTP service port on which CAPIF Ingress Gateway is exposed	
`global.publicHttpsSignallingPort`	HTTPS service port on which CAPIF Ingress Gateway is exposed	

Table 4-8 (Cont.) External Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`enableIncomingHttp`	This flag is for enabling/disabling HTTP/2.0 (insecure) in Ingress Gateway.	If the value is set to false, EG will not accept any HTTP/2.0 (insecure) Traffic. **Note:** This parameter is applicable only for the External Ingress GW.
`enableIncomingHttps`	This flag is for enabling/disabling HTTPS/2.0 (secure) in Ingress Gateway.	**Note:** This parameter is applicable only for the External Ingress GW. If the value is set to false, EG will not accept any HTTPS/2.0 (secure) Traffic
`image.name`	Ingress Gateway image name	This is an optional parameter. **Default Value:** ocingress_gateway
`image.tag`	Ingress Gateway image tag	This is an optional parameter. **Default Value:** CAPIF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`initContainersImage.name`	Image Name for Ingress GW init container	This is an optional parameter. **Default Value:** configurationinit
`initContainersImage.tag`	Tag Name for Ingress Gateway init container	This is an optional parameter. **Default Value:** CAPIF Images
`initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`updateContainersImage.name`	Image Name for Ingress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
`updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** CAPIF Images

Table 4-8 (Cont.) External Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`ports.actuatorPort`		**Default Value:** 9090
`ports.containerPort`		**Default Value:** 8081
`ports.containersslPort`		**Default Value:** 8443
`jaegerTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing at Ingress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
`openTelemetry.jaeger.httpExporter.host`	Specifies the host name of Jaeger Agent service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger Agent service	**Default Value:** 4318
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5
`service.ssl.tlsVersion`	The TLS version.	
`service.ssl.privateKey.k8SecretName`	Secret name that contains CAPIF Ingress Gateway Private Key	
`service.ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	
`service.ssl.privateKey.rsa.filename`	CAPIF's Private Key (RSA type) file name	
`service.ssl.privateKey.ecdsa.filename`	CAPIF Ingress Gateway Private Key (ecdsa type) file name	
`service.certificate.k8SecretName`	Secret name that contains CAPIF Ingress Gateway certificate for HTTPS	
`service.certificate.k8NameSpace`	Namespace in which k8SecretName is present	
`service.certificate.rsa.filename`	CAPIF Ingress Gateway Certificate (RSA type) file name	
`service.certificate.ecdsa.filename`	CAPIF Ingress Gateway Certificate (ECDSA type) file name	
`service.caBundle.k8SecretName`	Secret name that contains CAPIF Ingress Gateway's CA details for HTTPS	
`service.caBundle.k8NameSpace`	Namespace that contains CAPIF Ingress Gateway's CA details for HTTPS	

Table 4-8 (Cont.) External Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`caBundle.filename`	CAPIF Ingress Gateway's CA bundle filename	
`service.keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	
`service.keyStorePassword.k8NameSpace`	Namespace in which CAPIF Ingress Gateway's keystore password is present	
`service.keyStorePassword.filename`	CAPIF Ingress Gateway's Key Store password Filename	
`service.trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	CAPIF Ingress Gateway's trustStorePassword Filename	O
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway deployment.	
`log.level.root`	Log level for root logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.ingress`	Log level for ingress logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE

Table 4-8 (Cont.) External Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`log.level.oauth`	Log level for oauth logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.updateContainer`	Log level for updateContainer logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.configclient`	Log level for configclient logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.hook`	Log level for hook logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.cncc.security`	Log level for CNC Console security	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`readinessProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`readinessProbe.periodSeconds`	Specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3

Table 4-8 (Cont.) External Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 10
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 15
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1

Table 4-8 (Cont.) External Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 80
`requestTimeOut`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2500 ms
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5

Network Ingress GW

Table 4-9 Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`global.type`	The service type that will be used for this deployment.	It is not recommended to change the service type. **Default Value:** LoadBalancer
`global.staticIpAddressEnabled`	Specifies if static load balancer IP needs to be set	**Default Value:** false
`global.staticIpAddress`	Static IP address assigned to the Load Balancer from the external load balancer IP pool.	
`global.staticNodePortEnabled`	Specifies if static node port needs to be set	**Default Value:** false

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`global.staticHttpNodePort`	Static HTTP Node Port	
`global.staticHttpsNodePort`	Static HTTPS Node Port	
`global.publicHttpSignalingPort`	HTTP service port on which CAPIF Ingress Gateway is exposed	
`global.publicHttpsSignallingPort`	HTTPS service port on which CAPIF Ingress Gateway is exposed	
`enabled`	Specifies if the service is enabled.	**Default Value:** true
`image.name`	Ingress Gateway image name	This is an optional parameter. **Default Value:** ocingress_gateway
`image.tag`	Ingress Gateway image tag	This is an optional parameter. **Default Value:** CAPIF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`initContainersImage.name`	Image Name for Ingress GW init container	This is an optional parameter. **Default Value:** configurationinit
`initContainersImage.tag`	Tag Name for Ingress Gateway init container	This is an optional parameter. **Default Value:** CAPIF Images
`initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`updateContainersImage.name`	Image Name for Ingress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
`updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** CAPIF Images

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`extraContainers`	Controls the usage of extra container (DEBUG tool).	**Default Value:** USE_GLOBAL_VALUE
`service.ssl.tlsVersion`	The TLS version.	
`service.ssl.privateKey.k8SecretName`	Secret name that contains CAPIF Ingress Gateway Private Key	
`service.ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	
`service.ssl.privateKey.rsa.filename`	CAPIF's Private Key (RSA type) file name	
`service.ssl.privateKey.ecdsa.filename`	CAPIF Ingress Gateway Private Key (ecdsa type) file name	
`service.certificate.k8SecretName`	Secret name that contains CAPIF Ingress Gateway certificate for HTTPS	
`service.certificate.k8NameSpace`	Namespace in which k8SecretName is present	
`service.certificate.rsa.filename`	CAPIF Ingress Gateway Certificate (RSA type) file name	
`service.certificate.ecdsa.filename`	CAPIF Ingress Gateway Certificate (ECDSA type) file name	
`service.caBundle.k8SecretName`	Secret name that contains CAPIF Ingress Gateway's CA details for HTTPS	
`service.caBundle.k8NameSpace`	Namespace that contains CAPIF Ingress Gateway's CA details for HTTPS	
`caBundle.filename`	CAPIF Ingress Gateway's CA bundle filename	
`service.keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	
`service.keyStorePassword.k8NameSpace`	Namespace in which CAPIF Ingress Gateway's keystore password is present	
`service.keyStorePassword.filename`	CAPIF Ingress Gateway's Key Store password Filename	
`service.trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	CAPIF Ingress Gateway's trustStorePassword Filename	O

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway deployment.	
`ports.containerPort`	The container port detail.	**Default Value:** 8081
`ports.containersslPort`	The SSL container port detail.	**Default Value:** 8443
`ports.actuatorPort`	The actuator port detail.	**Default Value:** 9090
`log.level.root`	Log level for root logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.ingress`	Log level for ingress logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.oauth`	Log level for oauth logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`log.level.updateContainer`	Log level for updateContainer logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.configclient`	Log level for configclient logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.hook`	Log level for hook logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.cncc.security`	Log level for CNC Console security	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.traceIdGenerationEnabled`		**Default Value:** true
`startupProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`startupProbe.periodSeconds`	Specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`startupProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 10
`startupProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`startupProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 6
`readinessProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`readinessProbe.periodSeconds`	Specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 10
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 15
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 80
`requestTimeOut`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2500 ms
`minAvailable`	Number of Pods must always be available, even during a disruption.	**Default Value:** 2
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5
`jaegerTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing at Ingress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
`openTelemetry.jaeger.httpExporter.host`	Specifies the host name of Jaeger Agent service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger Agent service	**Default Value:** 4318

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5
`oauthValidatorEnabled`	Specifies if the OAuth validator must be enabled.	
`nfType`	NFType of service consumer.	
`nfInstanceId`	NF InstanceId of service consumer.	
`producerScope`	A comma-seperated list of services hosted by the service producer.	This is a mandatory field if `oauthValidatorEnabled` is set to **true**.
`allowedClockSkewSeconds`	Set this value if clock on the parsing NF(producer) is not perfectly in sync with the clock on the NF(consumer) that created the JWT.	This is a mandatory field if `oauthValidatorEnabled` is set to **true**.
`enableInstanceIdConfigHook`		
`nrfPublicKeyKubeSecret`	It consist of the name of the secret that stores the NRF publicKey.	
`nrfPublicKeyKubeNamespace`	It consist of the name of the namespace of the NRF publicKey secret.	
`validationType`	The validation type.	Possible values: • strict: the incoming request without "Authorization"(Access Token) header will be rejected. • relaxed: if incoming request contains "Authorization" header, it will be validated This is a mandatory field if `oauthValidatorEnabled` is set to **true**.
`producerPlmnMNC`	It consist of the name of the MNC of the service producer.	
`producerPlmnMCC`	It consist of the name of the MCC of the service producer.	
`oauthErrorConfigForValidationFailure`	The error configurations for the OAuth validation failure.	
`oauthErrorConfigOnTokenAbsence`	The error configurations for the absense of token failure.	
`initssl`		
`enableIncomingHttp`	This flag is for enabling/disabling HTTP/2.0 (insecure TLS) in Egress Gateway.	

Table 4-9 (Cont.) Network Ingress Gateway (`ingressgateway`) Parameters

Parameter	Description	Details
`enableIncomingHttps`	This flag is for enabling/disabling HTTPS/2.0 (secure TLS) in Egress Gateway.	
`enableOutgoingHttps`	This flag is for enabling/disabling HTTPS/2.0 (secured TLS) in Egress Gateway.	
`serviceMeshCheck`	Specifiies if Service Mesh would be present where NEF is deployed	
`isSbiTimerEnabled`	Specifies if SBI Timer is enabled	
`autoRedirect`	Allows Ingress GW to redirect to the URI present in the location header	
`metricPrefix`	Prefix to be added to all the metrics in the ingress gateway	
`metricSuffix`	Suffix to be added to all the metrics in the ingress gateway	
`pingDelay`	Delay between pings in seconds	
`ingressGwCertReloadEnabled`	Specifies if Ingress GW certificate can be reloaded	
`ingressGwCertReloadPath`	Certificate reload path	
`maxConcurrentPushedStreams`		**Default value:** 1000
`maxRequestsQueuedPerDestination`		
`maxConnectionsPerDestination`		Applicable if `serviceMeshCheck` is enabled.
`maxConnectionsPerIp`		Applicable if `serviceMeshCheck` is enabled.
`connectionTimeout`		Applicable if `serviceMeshCheck` is enabled.
`requestTimeout`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	Applicable if `serviceMeshCheck` is enabled.
`jettyGracefulRequestTermination`		Applicable if `serviceMeshCheck` is enabled.

4.2.1.6 Egress Gateway Parameters

The following table describes the parameters for the External Egress GW and Network Egress GW services.

Network Egress GW

Table 4-10 External Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`serviceEgressGateway.port`		
`deploymentEgressGateway.image.name`	Egress Gateway image name	This is an optional parameter. **Default Value:** ocegress_gateway
`deploymentEgressGateway.image.tag`	Egress Gateway image tag	This is an optional parameter. **Default Value:** Table 4-2
`deploymentEgressGateway.image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`initContainersImage.name`	Image Name for Egress Gateway init container	This is an optional parameter. **Default Value:** configurationinit
`initContainersImage.tag`	Tag Name for Egress Gateway init container	This is an optional parameter. **Default Value:** CAPIF Images
`initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`updateContainersImage.name`	Image Name for Egress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
`updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** Table 5-2

Table 4-10 (Cont.) External Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`extraContainers`	Specifies if extra container must be used for DEBUG tool.	
`initssl`		This value must always be set as true. **Default Value:** true
`enableOutgoingHttps`	This flag is for enabling/disabling HTTPS/2.0 (secured TLS) in Egress Gateway.	
`pingDelay`	Delay between pings in seconds.When set to <=0,ping is disabled	
`startupProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the startup probe	
`startupProbe.periodSeconds`	specifies that the kubelet should perform a startupProbe probe every xx seconds	
`startupProbe.timeoutSeconds`		
`startupProbe.successThreshold`		
`startupProbe.failureThreshold`		
`readinessProbe.initialDelaySeconds`		
`readinessProbe.periodSeconds`	specifies that the kubelet should perform a readinessProbe probe every xx seconds	
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	

Table 4-10 (Cont.) External Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	
`log.level.root`	Log level for root logs	
`log.level.egress`	Log level for egress logs	
`log.level.oauth`	Log level for oauth logs	
`log.level.updateContainer`	Log level for updateContainer logs	
`log.level.hook`	Log level for hook logs	
`service.ssl.keyType`	The selected key type.	
`service.ssl.tlsVersion`	The TLS version.	
`service.ssl.privateKey.k8SecretName`	Secret name that contains NEF Ingress Gateway Private Key	**Default Value:** TLSv1.2
`service.ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	
`service.ssl.privateKey.rsa.filename`	NEF's Private Key (RSA type) file name	
`service.ssl.privateKey.ecdsa.filename`	NEF Egress Gateway Private Key (ecdsa type) file name	
`service.certificate.k8SecretName`	Secret name that contains NEF Egress Gateway certificate for HTTPS	
`service.certificate.k8NameSpace`	Namespace in which k8SecretName is present	
`service.certificate.rsa.filename`	NEF Egress Gateway Certificate (RSA type) file name	
`service.certificate.ecdsa.filename`	NEF Egress Gateway Certificate (ECDSA type) file name	
`service.caBundle.k8SecretName`	Secret name that contains NEF Egress Gateway's CA details for HTTPS	
`service.caBundle.k8NameSpace`	Namespace that contains NEF Egress Gateway's CA details for HTTPS	
`expgw-apirouter.caBundle.filename`	NEF Egress Gateway's CA bundle filename	
`service.keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	
`service.keyStorePassword.k8NameSpace`	Namespace in which NEF Egress Gateway's keystore password is present	

Table 4-10 (Cont.) External Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`service.keyStorePassword.filename`	NEF Egress Gateway's Key Store password Filename	
`service.trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	NEF Egress Gateway's trustStorePassword Filename	O
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Egress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Egress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway deployment.	
`globalRemoveRequestHeader`	Attribute for blocklisting (removing) a request header at global level.	
`globalRemoveResponseHeader`	Attribute for blocklisting (removing) a response header at global level.	
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.commonHooksCpu`	Maximum amount of common hook CPU that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi

Table 4-10 (Cont.) External Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`resources.limits.commonHooksMemory`	Maximum amount of hook service memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.commonHooksCpu`	The amount of hook service CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.commonHooksMemory`	The amount of hook service memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 60
`minAvailable`	Minimum available pods	**Default Value:** 2
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5
`jettyIdleTimeout`		

Table 4-10 (Cont.) External Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`metricPrefix`		
`metricSuffix`		
`autoRedirect`		
`isSbiTimerEnabled`		
`sbiTimerTimezone`		
`egressGwCertReloadEnabled`		
`egressGwCertReloadPath`		
`jaegerTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing at Engress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
`openTelemetry.jaeger.httpExporter.host`	Specifies the host of Jaeger collector service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger collector service	**Default Value:** 4318
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5

Network Egress GW

Table 4-11 Network Egress Gateway (`egressgateway`) Parameters

Parameter	Description	Details
`cmName`		**Default Value:** egressgateway
`prefix`		**Default Value:** network
`serviceEgressGateway.port`		
`serviceEgressGateway.sslPort`		
`serviceEgressGateway.actuatorPort`		
`deploymentEgressGateway.image.name`	Egress Gateway image name	This is an optional parameter. **Default Value:** ocegress_gateway
`deploymentEgressGateway.image.tag`	Egress Gateway image tag	This is an optional parameter. **Default Value:** CAPIF Images

Table 4-11 (Cont.) Network Egress Gateway (egressgateway) Parameters

Parameter	Description	Details
deploymentEgressGateway.image.pullPolicy	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
initContainersImage.name	Image Name for Egress Gateway init container	This is an optional parameter. **Default Value:** configurationinit
initContainersImage.tag	Tag Name for Egress Gateway init container	This is an optional parameter. **Default Value:** CAPIF Images
initContainersImage.pullPolicy	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
updateContainersImage.name	Image Name for Egress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
updateContainersImage.tag	Tag Name for update container	This is an optional parameter. **Default Value:** NEF Images
updateContainersImage.pullPolicy	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
extraContainers	Specifies if extra container must be used for DEBUG tool.	
initssl		This value must always be set as true. **Default Value:** true
enableOutgoingHttps	This flag is for enabling/disabling HTTPS/2.0 (secured TLS) in Egress Gateway.	
pingDelay	Delay between pings in seconds.When set to <=0,ping is disabled	
httpsTargetOnly	Select SCP instances for https list only	
httpRuriOnly		

Table 4-11 (Cont.) Network Egress Gateway (egressgateway) Parameters

Parameter	Description	Details
startupProbe.initialDelaySeconds	Tells the kubelet that it should wait xx second before performing the startup probe	
startupProbe.periodSeconds	specifies that the kubelet should perform a startupProbe probe every xx seconds	
startupProbe.timeoutSeconds		
startupProbe.successThreshold		
startupProbe.failureThreshold		
readinessProbe.initialDelaySeconds		
readinessProbe.periodSeconds	specifies that the kubelet should perform a readinessProbe probe every xx seconds	
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	
livenessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	
livenessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	
K8ServiceCheck		
configureDefaultRoute	Flag to configure default route in Egress Gateway. Configure this flag when sbiRoutingConfigMode and routeConfigMode are configured as REST	
sbiRoutingConfigMode	Mode of operation for sbiRouting. Possible values are HELM, REST	

Table 4-11 (Cont.) Network Egress Gateway (egressgateway) Parameters

Parameter	Description	Details
routeConfigMode	Mode of configuration for configuring routes. Possible values are HELM, REST	
log.level.root	Log level for root logs	
log.level.egress	Log level for egress logs	
log.level.oauth	Log level for oauth logs	
log.level.updateContainer	Log level for updateContainer logs	
log.level.hook	Log level for hook logs	
service.ssl.keyType	The selected key type.	
service.ssl.tlsVersion	The TLS version.	
service.ssl.privateKey.k8SecretName	Secret name that contains NEF Ingress Gateway Private Key	**Default Value:** TLSv1.2
service.ssl.privateKey.k8NameSpace	Namespace in which k8SecretName is present	
service.ssl.privateKey.rsa.filename	NEF's Private Key (RSA type) file name	
service.ssl.privateKey.ecdsa.filename	NEF Egress Gateway Private Key (ecdsa type) file name	
service.certificate.k8SecretName	Secret name that contains NEF Egress Gateway certificate for HTTPS	
service.certificate.k8NameSpace	Namespace in which k8SecretName is present	
service.certificate.rsa.filename	NEF Egress Gateway Certificate (RSA type) file name	
service.certificate.ecdsa.filename	NEF Egress Gateway Certificate (ECDSA type) file name	
service.caBundle.k8SecretName	Secret name that contains NEF Egress Gateway's CA details for HTTPS	
service.caBundle.k8NameSpace	Namespace that contains NEF Egress Gateway's CA details for HTTPS	
expgw-apirouter.caBundle.filename	NEF Egress Gateway's CA bundle filename	
service.keyStorePassword.k8SecretName	Secret name that contains keyStorePassword	
service.keyStorePassword.k8NameSpace	Namespace in which NEF Egress Gateway's keystore password is present	
service.keyStorePassword.filename	NEF Egress Gateway's Key Store password Filename	
service.trustStorePassword.k8SecretName	Secret name that contains trustStorePassword	

Table 4-11 (Cont.) Network Egress Gateway (`egressgateway`) Parameters

Parameter	Description	Details
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	NEF Egress Gateway's trustStorePassword Filename	O
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Egress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Egress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway deployment.	
`globalRemoveRequestHeader`	Attribute for blocklisting (removing) a request header at global level.	
`globalRemoveResponseHeader`	Attribute for blocklisting (removing) a response header at global level.	
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.commonHooksCpu`	Maximum amount of common hook CPU that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.commonHooksMemory`	Maximum amount of hook service memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi

Table 4-11 (Cont.) Network Egress Gateway (`egressgateway`) Parameters

Parameter	Description	Details
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.commonHooksCpu`	The amount of hook service CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.commonHooksMemory`	The amount of hook service memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 60
`minAvailable`	Minimum available pods	**Default Value:** 2
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5
`cipherSuites`	CipherSuites for TLS1.2	
`allowedCipherSuites`	Allowed CipherSuites for TLS1.2	
`oauthClient.enabled`	Flag to enable accessToken request through Egress Gateway	

Table 4-11 (Cont.) Network Egress Gateway (egressgateway) Parameters

Parameter	Description	Details
nrfClientQueryEnabled	Determines if NRF-Client Query is enabled or not (Dynamic configuration).	
subscriptionRetryScheduledDelay		
httpsEnabled	Determines if https support is enabled or not which is a deciding factor for oauth request scheme.	
staticNrfList	List of Static NRF instances that need to be used for oAuth requests when nrfClientQueryEnabled is false.	
nfType	NFType of service consumer.	
nfInstanceId	NF InstanceId of service consumer.	
consumerPlmnMNC	MNC of service Consumer	
consumerPlmnMCC	MCC of service Consumer	
maxNonPrimaryNrfs		
apiPrefix		
retryErrorCodeSeriesForSameNrf		
retryErrorCodeSeriesForNextNrf		
retryExceptionListForSameNrf		
retryExceptionListForNextNrf		
connectionTimeout		
requestTimeout	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2500 ms
attemptsForPrimaryNRFInstance		
attemptsForNonPrimaryNRFInstance		
defaultNRFInstance		
defaultErrorCode		
nrfClientConfig.serviceName		
nrfClientConfig.host		
nrfClientConfig.port		
nrfClientConfig.nrfClientRequestMap		

Table 4-11 (Cont.) Network Egress Gateway (egressgateway) Parameters

Parameter	Description	Details
headerIndexing.doNotIndex		
requestTimeout	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2500 ms
metricPrefix		
metricSuffix		
jaegerTracingEnabled	Specifies whether to enable or disable Jaeger Tracing at Engress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
openTelemetry.jaeger.httpExporter.host	Specifies the host name of Jaeger Agent service	**Default Value:** jaeger-collector.cne-infra
openTelemetry.jaeger.httpExporter.port	Specifies the port of Jaeger Agent service	**Default Value:** 4318
openTelemetry.jaeger.probabilisticSampler	Specifies the Jaeger message sampler	**Default Value:** 0.5
tolerations	Specifies whether to enable or disable Jaeger Tracing at Engress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
helmBasedConfigurationNodeSelectorApiVersion		
nodeSelector		
nodeKey		
nodeValue		

4.3 Upgrading CAPIF

This section provides information about upgrading CAPIF deployment to the latest release.

> **Note**
>
> In a georedundant deployment, perform the steps explained in this section on all the georedundant sites.

4.3.1 Supported Upgrade Paths

The following table lists the supported upgrade paths for CAPIF.

Table 4-12 Supported Upgrade Paths

Source CAPIF release	Target CAPIF release
24.1.x	24.2.2
23.4.x	24.2.2

> **Note**
>
> CAPIF must be upgraded before upgrading NEF.

4.3.2 Preupgrade Tasks

This section provides information about preupgrade tasks to be performed before upgrading CAPIF.

- Keep current `oc-capif-<current release>-custom-values.yaml` file as backup.
- Update the new `oc-capif-<new release>-custom-values.yaml` file for target CAPIF release. For details on customizing this file, see Customizing CAPIF.
- Before starting the upgrade, take a manual backup of CAPIF REST based configuration. This helps if preupgrade data has to be restored.

  > **Note**
  >
  > For Rest API configuration details, see *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

- Install or upgrade the network policies, if applicable. For more information, see Configuring Network Policies for CAPIF.
- Before upgrading, perform sanity check using Helm test. See Performing Helm Test section for the Helm test procedure.

4.3.3 Upgrade Tasks

This section provides information about the sequence of tasks to be performed for upgrading an existing CAPIF deployment.

> **Note**
>
> It is recommended to perform CAPIF upgrade in a specific order. For more information about the upgrade order, see *Oracle Communications Cloud Native Core, Solution Upgrade Guide*.

When you attempt to upgrade an existing CAPIF deployment, the running set of containers and pods are replaced with the new set of containers and pods. However, If there is no change in the pod configuration, the running set of containers and pods are not replaced.

If you need to change any configuration then change the oc-capif-24.2.2-custom-values.yaml file with new values.

> **Note**
>
> It is advisable to create backup of the file before changing any configuration.

To configure the parameters, see Customizing CAPIF.

Execute the following command to upgrade an existing CAPIF deployment:

```
helm upgrade <release> <chart> -f oc-capif-24.2.2-custom-values.yaml
```

Table 4-13 Parameters and Definitions during CAPIF Upgrade

Parameters	Definitions
<chart>	It is the name of the chart that is of the form <repository/occapif>. For example: reg-1/occapif or cne-repo/occapif
<release>	It can be found in the output of helm list command

In case of backout:

1. Check the history of helm deployment:
   ```
   helm history <helm_release>
   ```
2. Rollback to the required revision:
   ```
   helm rollback <release name> <revision number>
   ```

> **Note**
>
> Perform sanity check using Helm test. See Performing Helm Test section for the Helm test procedure.

4.4 Rolling Back CAPIF

This section provides information about rolling back CAPIF deployment to the previous release.

Note

- In a georedundant deployment, perform the steps explained in this section on all georedundant sites separately.
- For upgrade or rollback, a default timeout of 5 minutes is provided by Helm. In case of a slow server, it might take more time. In such scenarios, the upgrade status will be `pending-upgrade` or `pending-rollback`.
  To avoid this, add `--timeout` flag according to their network speed.

  Example

  Upgrade:

  ```
  helm upgrade oncef oc-nef -f experiment.yaml –-timeout=45m -n mv --
  dry-run
  ```

  Rollback:

  ```
  helm rollback oncef 1 -n mv –-timeout=45m
  ```

4.4.1 Supported Rollback Paths

The following table lists the supported rollback paths for CAPIF:

Table 4-14 Supported Rollback Path

Source Release	Target Release
24.2.2	24.1.x
24.2.2	23.4.x

Note

If georedundancy feature was disabled before upgrading to 23.1.x, then rolling back to a previous version will automatically disable this feature. However, the database will still have records of the `NfInstances` and `NfSubscriptions` from the mated sites. For more information, contact My Oracle Support.

4.4.2 Rollback Tasks

To roll back from CAPIF 24.1.x to a previous version:

> **⚠ Caution**
>
> - No configuration should be performed during rollback.
> - Do not exit from `helm rollback` command manually. After running the `helm rollback` command, it takes some time (depending upon number of pods to rollback) to rollback all of the services. In the meantime, you must not press "**ctrl+c**" to come out from `helm rollback` command. It may lead to anomalous behavior.

1. Run the following command to check the revision you must roll back to:

   ```
   $ helm history <release_name> -n <release_namespace>
   ```

2. Run the command to rollback to the required revision:

   ```
   $ helm rollback <release_name> <revision_number> -n <release_namespace>
   ```

> **ⓘ Note**
>
> If the rollback is not successful, perform the troubleshooting steps mentioned in *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

4.5 Uninstalling CAPIF

This section provides information about uninstalling CAPIF.

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

4.5.1 Uninstalling CAPIF Using Helm

Prerequisite: Ensure to uninstall NEF before uninstalling CAPIF. For more information about uninstalling NEF, see Uninstalling NEF.

To uninstall CAPIF, run the following command:

Run the following command to completely delete or remove the CAPIF deployment:

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

where,

release_name is a name provided to identify the helm deployment.

release-namespace is the name provided to identify the namespace of CAPIF deployment.

For example:

```
helm uninstall occapif -n occapif
```

Helm keeps a record of its releases, so you can still reactivate the release after uninstalling it.

To completely remove a release from the cluster, add the `--purge` parameter to `helm delete` command:

```
helm delete --purge release_name
```

For example:

```
helm delete --purge occapif
```

To verify the CAPIF uninstallation status of the pods, run the following command:

```
kubectl get pods -<release-namespace>
```

Note

- During helm uninstallation if the Kubernetes jobs started by the helm hooks get stuck, then you can use the following command to delete the jobs manually:

  ```
  while true; do kubectl delete jobs --all -n <release-namespace>;
  sleep 5;done
  ```

- If the command output displays the CAPIF resources or objects, then perform Deleting Kubernetes Resources.
  If the command output displays any Kubernetes namespace, then perform the Deleting Kubernetes Namespace.

4.5.2 Deleting Kubernetes Namespace

This section describes how to delete Kubernetes namespace where CAPIF is deployed.

Run the following command to delete the Kubernetes namespace:

Note

Be sure before removing the namespace as it deletes all the resources or objects created in the namespace.

```
kubectl delete namespace <occapif Kubernetes namespace>
```

For example:

```
kubectl delete namespace occapif
```

To verify the CAPIF uninstallation, run the following command:

```
$ kubectl get all -n <release-namespace>
```

In case of successful uninstallation, no CAPIF resource is displayed in the command output.

If the command output displays the CAPIF resources or objects, then perform the following procedure:

1. Run the following command to delete all the objects:
   - To delete all the Kubernetes objects:

     ```
     kubectl delete all --all -n <release-namespace>
     ```

   - To delete all the configmaps:

     ```
     kubectl delete cm --all -n <release-namespace>
     ```

   Caution

   The command deletes all the Kubernetes objects of the specified namespace. In case you have created the RBAC resources and service accounts before the helm installation in the same namespace, and these resources are required, then do not delete them.

2. Run the following command to delete the specific resources:

   ```
   kubectl delete <resource-type> <resource-name> -n <release-namespace>
   ```

3. Run the following command to delete the Kubernetes namespace:

   ```
   kubectl delete namespace <release-namespace>
   ```

4.5.3 Deleting Kubernetes Resources

This section describes how to delete Kubernetes resources or objects where CAPIF is deployed.

Note

Be sure before running the following commands as it deletes all the Kubernetes objects in the specified namespace.

1. Run the following command to delete all the Kubernetes objects:

   ```
   kubectl delete all --all -n <release-namespace>
   ```

2. Run the following command to delete all the configmaps:

   ```
   kubectl delete cm --all -n <release-namespace>
   ```

3. Run the following command to delete the specific resources:

   ```
   kubectl delete <resource-type> <resource-name> -n <release-namespace>
   ```

4.5.4 Deleting the MySQL Details

Procedure for complete removal of MySQL database and username

This section describes how to complete removal of MySQL database and users.

To remove MySQL users while uninstalling CAPIF, run the following commands:

1. Log in to the machine which has permission to access the SQL nodes of NDB cluster.
2. Connect to the SQL node of NDB cluster successively.
3. Log in to the MySQL prompt using root permission or user, which has permission to drop the tables. For example:

   ```
   mysql -h 127.0.0.1 -uroot -p
   ```

   > **Note**
   >
   > This command may vary from system to system, path for MySQL binary, root user and root password. After running this command, the user must enter the password specific to the user mentioned in the command.

4. Run the following command to clean up the database:

   ```
   $ DROP DATABASE if exists <database name>
   ```

   For example, to remove a database named **occapif_db**, run the following command:

   ```
   DROP DATABASE IF EXISTS occapif_db;
   ```

5. Run the following command to remove the NRF MySQL Users:
   Remove CAPIF privileged user:

   ```
   $ DROP USER IF EXISTS <CAPIF Privileged-User Name>;
   ```

   Example:

   ```
   $ DROP USER IF EXISTS 'priviledgeduser@'%';
   ```

   Remove CAPIF application user:

   ```
   $ DROP USER IF EXISTS <CAPIF Application User Name>;
   ```

   Example:

   ```
   $ DROP USER IF EXISTS 'appuser@'%';
   ```

> ⚠ **Caution**
>
> Removal of users must be done on all the SQL nodes for all CAPIF sites.

6. Exit from MySQL prompt and SQL node.

5
NEF Installation and Upgrade

This chapter describes how to install, customize, and uninstall Network Exposure Function (NEF) on Cloud Native Environment (CNE).

5.1 Installing NEF

This section provides information about installing NEF in a cloud native environment.

> **Note**
>
> NEF supports fresh installation, and it can also be upgraded from 23.3.x and 23.4.x. For information about the prerequisites to install NEF, see Prerequisites chapter and to know how to upgrade NEF, see Upgrading NEF section.

5.1.1 Preinstallation

Before installing NEF, perform the tasks described in this section.

5.1.1.1 Verifying and Creating NEF Namespace

This section explains how to verify or create a new namespace in the system.

> **Note**
>
> This is a mandatory procedure, run this before proceeding further with the installation procedure. The namespace created or verified in this procedure is an input for the next procedures.

To verify if the required namespace already exists in the system, run the following command:

```
$ kubectl get namespaces
```

In the output of the above command, check if the required namespace is available. If not available, create the namespace using the following command:

> **Note**
>
> This is an optional step. Skip this step if the required namespace already exists.

```
$ kubectl create namespace <required namespace>
```

For example:

```
$ kubectl create namespace ocnef-namespace
```

Naming Convention for Namespaces

The namespace should:

- start and end with an alphanumeric character.
- contain 63 characters or less.
- contain only alphanumeric characters or '-'.

> **Note**
>
> It is recommended to avoid using prefix `kube-` when creating namespace as this prefix is reserved for Kubernetes system namespaces.

5.1.1.2 Creating Service Account, Role, and RoleBinding

This section describes the procedure to create a service account, role, and role binding resources. The Secret(s) can be under same namespace where NEF is getting deployed (recommended), or you can select to use different namespaces for different secret(s). If all the secret(s) are under the same namespace as NEF, then you can bind the Kubernetes Role with the given ServiceAccount. Otherwise, it is required to bind the ClusterRole with the given ServiceAccount.

1. Create an NEF resource file:

   ```
   vi <ocnef-resource-file>
   ```

   For example:

   ```
   vi ocnef-resource-template.yaml
   ```

2. Update the `ocnef-resource-template.yaml` with release specific information:

   > **Note**
   >
   > Update <helm-release> and <namespace> with its respective NEF namespace and NEF helm release name.

   ```
   ## Sample template start#
   apiVersion: v1
   kind: ServiceAccount
   metadata:
     name: <helm-release>-ocnef-serviceaccount
     namespace: <namespace>
   ---

   apiVersion: rbac.authorization.k8s.io/v1
   kind: Role
   ```

```
metadata:
  name: <helm-release>-ocnef-role
  namespace: <namespace>
rules:
- apiGroups:
  - "" # "" indicates the core API group
  resources:
  - services
  - configmaps
  - pods
  - secrets
  - endpoints
  verbs:
  - get
  - watch
  - list
---

apiVersion: rbac.authorization.k8s.io/v1beta1
kind: RoleBinding
metadata:
  name: <helm-release>-ocnef-rolebinding
  namespace: <namespace>
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: <helm-release>-ocnef-role
subjects:
- kind: ServiceAccount
  name: default
  namespace: <namespace>
## Sample template end#
```

3. Run the following command to create service account, role, and role binding:

   ```
   kubectl -n <ocnef-namespace> create -f ocnef-resource-template.yaml
   ```

   For example:

   ```
   kubectl -n ocnef create -f ocnef-resource-template.yaml
   ```

4. Update the `serviceAccountName` parameter in the `oc-nef-24.2.2-custom-values.yaml` file with the value updated in `name` field under `kind: ServiceAccount`. For more information about `serviceAccountName` parameter, see the Global Parameters section.

> **Note**
>
> The service account name configured in this section must be used as the value for the `serviceAccountName` parameter during the customization using the NEF Custom Values YAML file. For more information about the parameter, see Global Parameters.

5.1.1.3 Configuring Database, Creating Users, and Granting Permissions

NEF microservices use MySQL database to store the configuration and run time data.

NEF requires the database administrator to create a user in MySQL database and provide the necessary permissions to access the databases. Before installing NEF it is required to create the MySQL user and databases.

> **Note**
>
> While performing a fresh installation, in case the NEF release is already deployed, then purge the deployment and remove the databases and users used for the previous deployment. For uninstallation procedure, see Uninstalling NEF.

NEF Users

There are two types of NEF database users with a different set of permissions:

1. **NEF privileged user:** This user has a complete set of permissions. This user can create or delete the database and perform create, alter, or drop operations on the database tables for performing installation, upgrade, rollback, and delete operations.
2. **NEF application user:** This user has a limited set of permissions and is used by NEF application during service operations handling. This user can insert, update, get, and remove the records. This user cannot create, alter, and drop the database and the tables.

> **Note**
>
> If cnDBTier 24.2.2 is used during installation, set the `ndb_allow_copying_alter_table` parameter to `'ON'` in the `ocnef_dbtier_24.2.2_custom_values_24.2.2.yaml` file before installing NEF. After NEF installation, set the parameter to its default value `'OFF'`.

5.1.1.3.1 Single Site

This section explains how a database administrator can create database and users for a single site deployment.

> **Note**
>
> New MySQL users along with their privileges must be added manually in each SQL node, of cnDBtier namespace, for NEF site.

1. Log in to the machine where ssh keys are stored. The machine must have permission to access the SQL nodes of NDB cluster.
2. Connect to the SQL nodes.
3. Log in to the database as a root user.

4. Create the NEF Release database:

```
CREATE DATABASE <ocnef_release_database_name>;
```

For example:

```
CREATE DATABASE ocnef_releaseDb ;
```

> **Note**
>
> In case of georedundant deployment, each NEF site must have different Release database names.

5. Create the NEF Service database:

```
CREATE DATABASE <ocnef_service_database_name>;
```

For example:

```
CREATE DATABASE ocnef_db;
```

> **Note**
>
> In case of georedundant deployment, each NEF site must have same Service database name.

6. Create a privileged user and grant all the necessary permissions to the user.
   a. Run the following command to create privileged user:

```
CREATE USER '<ocnef privileged username>'@'%' IDENTIFIED BY '<ocnef
privileged user password>';
```

> **Note**
>
> If MySQL is 8.0, run the following command to create a user:
>
> ```
> CREATE USER IF NOT EXISTS '<ocnef privileged username>'@'%'
> IDENTIFIED WITH mysql_native_password BY '<ocnef privileged user
> password>'
> ```

where:
<ocnef privileged username> is the username and *<ocnef privileged user password>* is the password for MySQL privileged user.

b. Run the following command to grant the necessary permissions to the privileged user:

```
GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES,
CREATE TEMPORARY TABLES, DELETE, UPDATE,
EXECUTE, NDB_STORED_USER , REFERENCES ON *.* TO '<ocnef privileged
username>'@'%';
```

For example:

In the following example "ocnefprivilegedusr" is used as username and "ocnefprivilegedpasswd" is used as the password. All the permissions are granted to the privileged user, that is, ocnefprivilegedusr.

```
CREATE USER 'ocnefprivilegedusr'@'%' IDENTIFIED BY 'ocnefprivilegedpasswd';



GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES,
CREATE TEMPORARY TABLES, DELETE, UPDATE,
EXECUTE, NDB_STORED_USER , REFERENCES ON *.* TO 'ocnefprivilegedusr'@'%';
```

7. Create an application user and grant all the necessary permissions.
   a. Run the following command to create application user:

      ```
      CREATE USER '<ocnef application username>'@'%' IDENTIFIED BY '<ocnef
      application user password>';
      ```

      where:

      <ocnef application username> is the username and *<ocnef application user password>* is the password for MySQL database user.

   b. Run the following command to grant the necessary permissions to the application user:

      ```
      GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES,
      CREATE TEMPORARY TABLES, DELETE, UPDATE,
      EXECUTE, NDB_STORED_USER , REFERENCES ON *.* TO '<ocnef application
      username>'@'%';
      ```

   For example:
   In the following example, "ocnefusr" is used as username and "ocnefpasswd" is used as its password. All the necessary permissions are granted to the application user, that is, ocnefusr. Here, default database names of microservices are used.

   ```
   CREATE USER 'ocnefusr'@'%' IDENTIFIED BY 'ocnefpasswd';



   GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES,
   CREATE TEMPORARY TABLES, DELETE, UPDATE,
   EXECUTE, NDB_STORED_USER , REFERENCES ON *.* TO 'ocnefusr'@'%';
   ```

> **Note**
>
> The database name is specified in the `dbName` parameter for NEF services in the `oc-nef-24.2.2-custom-values.yaml` file.

8. To confirm that the privileged or application user has all the permissions, run the following command:

   ```
   show grants for username;
   ```

   where, *username* is the privileged or application user's username.

   For example:

   ```
   show grants for ocnefprivilegedusr;


   show grants for ocnefusr;
   ```

9. Exit from database and log out from MySQL node.

5.1.1.3.2 Multisite

This section explains how database administrator can create the database and users for a multisite deployment.

> **Note**
>
> - Perform the steps in Single Site for creating database.
> - NEF supports only 2-site.

For information on the Database Configurations parameters, refer to Global Parameters section.

5.1.1.4 Configuring Secret for Enabling Access Token Validation

This section describes the procedure to configure Kubernetes secrets which will contain public certificate of CAPIF.

To create this secret, the public certificate of CAPIF should be transfered offline over mail or some other secured communication channels

> **Note**
>
> Public certificate is generated when installing CAPIF.

Create kubernetes secret:

1. Run the following command to create kubernetes secret:

```
$ kubectl create secret generic <certificate-secret> --from-file=<capif
public certificate> -n <Namespace of OCNEF deployment>
```

> **Note**
>
> Note down the command used during the creation of kubernetes secret. This command is used for updating the secrets in future.

Example:

```
$ kubectl create secret generic certificate-secret --from-file=cert.cer -n
ocnef
```

2. Verify the secret creation with the following command:

```
$ kubectl describe secret <certificate-secret> -n <Namespace of NEF
deployment>
```

Example:

```
$ kubectl describe secret certificate-secret -n ocnef
```

5.1.1.5 Configuring Kubernetes Secret for Accessing NEF Database

This section describes the procedure to configure Kubernetes secrets for accessing the databases created in the previous section.

5.1.1.5.1 Creating and Updating Kubernetes Secret for Accessing NEF Privileged Database User

This section explains the steps to create kubernetes secrets for accessing NEF database and privileged user details created by database administrator in the above section.

Create kubernetes secret for accessing the privileged user as follows:

1. Run the following command to create kubernetes secret:

```
$ kubectl create secret generic <privileged user secret name> --from-
literal=dbUsername=<OCNEF Privileged User Name> --from-
literal=dbPassword=<Password for OCNEF Privileged User>  --from-
literal=mysql-username=<OCNEF Privileged User Name> --from-literal=mysql-
password=<Password for OCNEF Privileged User> -n <Namespace of OCNEF
deployment>
```

> **Note**
>
> Note down the command used during the creation of kubernetes secret. This command is used for updating the secrets in future.

Example:

```
$ kubectl create secret generic privilegeduser-secret --from-
literal=dbUsername=ocnefPrivilegedUsr --from-
literal=dbPassword=ocnefPrivilegedPasswd  --from-literal=mysql-
username=ocnefPrivilegedUsr --from-literal=mysql-
password=ocnefPrivilegedPasswd -n ocnef_namespace
```

2. Verify the secret creation with the following command:

```
$ kubectl describe secret <privileged user secret name> -n <Namespace of
OCNEF deployment>
```

Example:

```
$ kubectl describe secret privilegeduser-secret -n ocnef_namespace
```

5.1.1.5.2 Creating and Updating Kubernetes Secret for Accessing NEF Application Database User

This section explains the steps to create secrets for accessing and configuring application database user created in above section.

Create kubernetes secret for NEF application user with the following steps:

1. Run the following command to create kubernetes secret:

```
$ kubectl create secret generic <appuser-secret name> --from-
literal=dbUsername=<OCNEF APPLICATION User Name> --from-
literal=dbPassword=<Password for OCNEF APPLICATION User>  --from-
literal=mysql-username=<OCNEF APPLICATION User Name> --from-literal=mysql-
password=<Password for OCNEF APPLICATION User> -n <Namespace of OCNEF
deployment>
```

> **Note**
>
> Note down the command used during the creation of kubernetes secret. This command is used for updating the secrets in future.

Example:

```
$ kubectl create secret generic appuser-secret --from-
literal=dbUsername=ocnefusr --from-literal=dbPassword=ocnefpasswd  --from-
literal=mysql-username=ocnefusr --from-literal=mysql-password=ocnefpasswd -
n ocnef_namespace
```

2. Verify the secret creation with the following command:

```
$ kubectl describe secret <appuser-secret name> -n <Namespace of OCNEF
deployment>
```

Example:

```
$ kubectl describe secret appuser-secret -n ocNEF
```

5.1.1.5.3 Creating and Updating Kubernetes Secret for Storing Security Certificates for NEF

To create a kubernetes secret for storing security certificates for NEF Exposure Gateway (EG), perform the following steps:

1. Run the following command to create kubernetes secret :

   ```
   $ kubectl create secret generic <ocnef-secret-name> --from-
   file=rsa_private_key_pkcs1.pem --from-file=trust.txt --fromfile=key.txt --
   from-file=ocegress.cer --from-file=caroot.cer -n<namespace>
   ```

   where,

   `<ocnef-secret-name>` can be `ext-expgw-secret` or `fivegc-service-secret`, as required

   rsa_private_key_pkcs1.pem is rsa private key

   trust.txt is trust store password

   key.txt is key store password

   caroot.cer is the cer chain for trust store

   cert.cer is signed server certificate

5.1.1.6 Configuring Secrets for Enabling HTTPS

This section describes the procedure to configure Kubernetes secrets for setting up secured communication of NEF with Ingress and Egress gateways.

> **Note**
>
> The passwords for TrustStore and KeyStore are stored in respective password files as mentioned in this section.

To create Kubernetes secret for HTTPS, following files are required:

- ECDSA private key and CA signed certificate of NEF, if initialAlgorithm is ES256
- RSA private key and CA signed certificate of NEF, if initialAlgorithm is RS256
- TrustStore password file
- KeyStore password file

> **Note**
>
> It is at the discretion of user to create the private keys and certificates, and it is not in the scope for NEF. This section lists only samples to create KeyPairs and certificates.

Update Secrets

This section explains how to update the secret with updated details.

1. Copy the exact command used in above section during creation of secret.
2. Update the same command with string "--dry-run -o yaml" and "kubectl replace -f - -n <Namespace of NEF deployment>".
   Example of the command syntax:

   ```
   $ kubectl create secret generic <ocnef-secret-name> --from-
   file=<ssl_ecdsa_private_key.pem> --from-file=<rsa_private_key_pkcs1.pem> --
   from-file=<ssl_truststore.txt> --from-file=<ssl_keystore.txt> --from-
   file=<caroot.cer> --from-file=<ssl_rsa_certificate.crt> --from-
   file=<ssl_ecdsa_certificate.crt> --dry-run -o yaml -n <Namespace of OCNEF
   deployment> | kubectl replace -f - -n <Namespace of OCNEF deployment>
   ```

   Example:

   The names used below are same as provided in `oc-nef-24.2.2-custom-values.yaml` in NEF deployment:

   ```
   $ kubectl create secret generic ocingress-secret --from-
   file=ssl_ecdsa_private_key.pem --from-file=rsa_private_key_pkcs1.pem --
   from-file=ssl_truststore.txt --from-file=ssl_keystore.txt --from-
   file=caroot.cer --from-file=ssl_rsa_certificate.crt --from-
   file=ssl_ecdsa_certificate.crt --dry-run -o yaml -n OCNEF | kubectl
   replace -f - -n OCNEF
   ```

3. Run the updated command.
   After successful secret update, the following message is displayed:

   ```
   secret/<ocingress-secret> replaced
   ```

5.1.1.7 Configuring Secrets to Enable Access Token

This section explains how to configure a secret for enabling access token.

5.1.1.7.1 Generating Private Keys and Certificates

This section describes the procedure to generate private key and certificate.

Create Private Keys and Certificates

1. Generate RSA private key by running the following command:

   ```
   openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout
   rsa_private_key -out rsa_certificate.crt
   ```

2. Convert the private key to `.pem` format with the following command:

   ```
   openssl rsa -in rsa_private_key -outform PEM -out rsa_private_key_pkcs1.pem
   ```

3. Generate cert out of private key, by running the following command:

   ```
   openssl req -new -key rsa_private_key -out tmp.csr -config ssl.conf
   ```

> **Note**
>
> The `ssl.conf` can be used to configure default entries along with SAN details for your cert.

The following snippet shows a sample of the `ssl.conf` syntax:

```
#ssl.conf
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
req_extensions = req_ext

[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = IN
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Karnataka
localityName = Locality Name (eg, city)
localityName_default = Bangalore
organizationName = Organization Name (eg, company)
organizationName_default = Oracle
commonName = Common Name (e.g. server FQDN or YOUR name)
commonName_max = 64
commonName_default = localhost

[ req_ext ]
subjectAltName = @alt_names

[alt_names]
IP = 127.0.0.1
DNS.1 = localhost
```

4. Create Root CA by running the following command:

```
openssl req -new -keyout cakey.pem -out careq.pem
openssl x509 -signkey cakey.pem -req -days 3650 -in careq.pem -out
caroot.cer -extensions v3_ca
echo 1234 > serial.txt
```

5. Sign server cert with root Ca private key, by running the following command:

```
openssl x509 -CA caroot.cer -CAkey cakey.pem -CAserial serial.txt -req -in
tmp.csr -out tmp.cer -days 365 -extfile ssl.conf -extensions req_ext
```

> **Note**
>
> The `ssl.conf` file must be reused, as SAN contents is not packaged when signing.

Import key and cert inside keystore

Steps to import key and cert inside the keystore manually:

1. Create keystore with default rsa keys, by running the following command:

   ```
   keytool -genkey -keyalg RSA -alias rsa -keystore keystore.jks
   ```

2. Delete the keys from the keystore to remove the default keys, by running the following command:

   ```
   keytool -delete -alias rsa -keystore keystore.jks
   ```

   Check if the keystore is empty with zero entries, by executing:

   ```
   keytool -list -v -keystore keystore.jks
   ```

3. Combine private key and cert in `pkcs12` format to import into keystore, by running the following command:

   ```
   openssl pkcs12 -export -out signedCert.pkcs12 -inkey rsa_private_key -in
   sample.cer
   ```

   > **Note**
   >
   > The certificate must be signed with the root Ca privatekey.

4. Import the `pkcs` file to the keystore, by running the following command:

   ```
   keytool -v -importkeystore -srckeystore signedCert.pkcs12 -srcstoretype
   PKCS12 -destkeystore keystore.jks
   -deststoretype JKS
   ```

5.1.1.8 Configuring Network Policies for NEF

Perform the following installation and upgrade steps for deploying network policies for NEF deployment.

> **Note**
>
> - Ports mentioned in policies are container ports and not the exposed service ports.
> - Connections that were created before installing network policy and still persist are not impacted by the new network policy. Only the new connections would be impacted.
> - If you are using ATS suite along with network policies, it is required to install the NEF, CAPIF, and ATS in the same namespace.
> - If the traffic is blocked or unblocked between the pods even after applying network policies, check if any existing policy is impacting the same pod or set of pods that might alter the overall cumulative behavior.
> - If changing default ports of services such as Prometheus, Database, Jaegar, or if Ingress or Egress Gateway names is overridden, update them in the corresponding network policies.
> - While using the debug container, it is recommended to uninstall the network policies or update them as required to establish the connections.

Configuring Network Policies

Following are the various operations that can be performed for network policies:

5.1.1.8.1 Installing Network Policy

Prerequisite

Network Policies are implemented by using the network plug-in. To use network policies, you must be using a networking solution that supports Network Policy.

> **Note**
>
> For a fresh installation, it is recommended to install Network Policies before installing NEF. However, if NEF is already installed, you can still install the Network Policies.

To install network policy

1. Open the `ocnef-network-policy-custom-values-24.2.2.yaml` file provided in the release package zip file. For downloading the file, see Installation Package Download.
2. The custom values are provided with the default security policies.
   If required, update the `ocnef-network-policy-custom-values-24.2.2.yaml` file as described in the Configurable Parameters of NEF.
3. Run the following command to install the network policy:

   ```
   helm install <helm-release-name> ocnef-network-policy/ -n <namespace> -f
   <custom-value-file>
   ```

   Sample command:
   ```
   helm install ocnef-network-policy ocnef-network-policy/ -n ocnef -f ocnef-
   network-policy-custom-values-24.2.2.yaml
   ```

   Where,

`helm-release-name`: ocnef-network-policy Helm release name

`custom-value-file`: ocnef-network-policy custom value file

`namespace`: namespace must be the NEF's namespace

5.1.1.8.2 Upgrading Network Policy

1. Modify the `ocnef-network-policy-custom-values-24.2.2.yaml` file to add new network policies or update the existing policies.
2. Run the following command to upgrade the network policy:

   ```
   helm upgrade <helm-release-name> ocnef-network-policy/ -n <namespace> -f
   <custom-value-file>
   ```

   Sample command:
   ```
   helm upgrade ocnef-network-policy ocnef-network-policy/ -n ocnef -f ocnef-
   network-policy-custom-values-24.2.2.yaml
   ```

   Where,

   `helm-release-name`: ocnef-network-policy Helm release name

   `custom-value-file`: ocnef-network-policy custom value file

   `namespace`: namespace must be the NEF's namespace

5.1.1.8.3 Verifying Network Policies

Run the following command to verify if the network policies are deployed successfully:

```
kubectl get <helm-release-name> -n <namespace>
```

For example:

```
kubectl get ocnef-network-policy -n ocnef
```

where,

- `helm-release-name`: ocnef-network-policy Helm release name.
- `namespace`: NEF namespace.

5.1.1.8.4 Uninstalling Network Policy

Run the following command to uninstall and remove all the network policy:

```
helm uninstall <helm-release-name> -n <namespace>
```

Sample command:

```
helm uninstall ocnef-network-policy -n ocnef
```

5.1.1.8.5 Configuration Parameters for Network Policies

Table 5-1 Configuration Parameters for Network Policy

Parameter	Description	Details

networkPolicies	The networkPolicies parameter is of array type. Each element of this array must have standard object's metadata and specification of the desired behavior for this NetworkPolicy.	This is an optional parameter. The network policy Helm chart creates policies for each entry.
		Example:
		`- metadata:`
		` name: <name of network policy>`
		`spec:`
		` <provide the standard specifications>`
		Note: Specify the policies compatible with apiversion networking.k8s.io/v1.

For more information about this functionality, see Network Policies in the *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

5.1.2 Installation Tasks

This section describes how to install NEF.

> **Note**
>
> Before installing NEF, you must complete Prerequisites and Preinstallation Tasks. In a georedundant deployment, perform the steps explained in this section on all georedundant sites.

5.1.2.1 Pushing the Images to Customer Docker Registry

NEF deployment package includes ready-to-use images and Helm charts to orchestrate containers in Kubernetes.

The following table lists the docker images for NEF:

Table 5-2 Docker Images for NEF

Service Name	Docker Image Name	Image Tag
Expiry Auditor Service	oc_nef_expiry_auditor	24.2.2
API Router	oc_nef_aef_apirouter	24.2.2
CCF Client	oc_nef_ccfclient_manager	24.2.2
Monitoring Events	oc_nef_monitoring_events	24.2.2
Quality of Service	oc_nef_quality_of_service	24.2.2

Table 5-2 (Cont.) Docker Images for NEF

Service Name	Docker Image Name	Image Tag
Traffic Influence	oc_nef_traffic_influence	24.2.2
Diameter Gateway	oc_nef_diam_gateway	24.2.2
Nrf Client Service	nrf-client	24.2.8
5GC Agent	oc_nef_5gcagent	24.2.2
APD Manager	oc_nef_apd_manager	24.2.2
Configuration Update Service	configurationupdate	24.2.15
Configuration INIT Service	configurationinit	24.2.15
Ingress Gateway	ocingress_gateway	24.2.15
Egress Gateway	ocegress_gateway	24.2.15
Common Config Hook	common_config_hook	24.2.15
Application Performance Service	oc-perf-info	24.2.15
Application Info Service	oc-app-info	24.2.15
Config-Server Service	oc-config-server	24.2.15
Debug Tools Service	ocdebug-tools	24.2.6
NF Test Service	nf_test	24.2.5
Device Trigger	oc_nef_device_trigger	24.2.2
Pool Manager	oc_nef_pool_manager	24.2.2
Camara	oc_camara_atf	24.2.2
Console Data Service	oc_nef_console_data_service	24.2.2
MSISDNLess MO SMS	oc_nef_msisdnless_mo_sms	24.2.2

Pushing Docker Images

Prerequisite: Download and untar the NEF Package ZIP file. For more information about downloading the package, see Installation Package Download.

To push the images to the registry:

1. Unzip the release package to the location where you want to install NEF. The package is as follows:
   `ocnef-pkg-24.2.2.0.0.tgz`
2. Untar the NEF package zip file to get NEF image tar file:

   ```
   tar -xvzf ocnef-pkg-24.2.2.0.0.tgz
   ```

   The directory consists of the following:

   - `ocnef-24.2.2.tgz`: Helm Chart
   - `ocnef-24.2.2.tgz.sha256`: Checksum for Helm chart tgz file
   - `ocnef-images-24.2.2.tar`: NEF Images File
   - `ocnef-images-24.2.2.tar.sha256`: Checksum for images tar file
   - `ocnef-network-policy-24.2.2.tgz`: Helm Chart for network policy
   - `ocnef-network-policy-24.2.2.tgz.sha256` Checksum for network policy Helm chart tgz file

3. Run one of the following commands to load the `ocnef-images-24.2.2.tar` file:

```
docker load --input /IMAGE_PATH/ocnef-images-24.2.2.tar
```

```
podman load --input /IMAGE_PATH/ocnef-images-24.2.2.tar
```

4. Run one of the following commands to verify the images are loaded:

```
docker images
```

```
podman images
```

> **Note**
>
> Verify the list of images shown in the output with the list of images shown in the above table. If the list does not match, reload the image tar files.

5. Run one of the following commands to tag the images to the registry:

```
docker tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
```

```
podman tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
```

6. Run one of the following commands to push the images to the registry:

```
docker push <docker-repo>/<image-name>:<image-tag>
```

```
podman push <docker-repo>/<image-name>:<image-tag>
```

> **Note**
>
> It is recommended to configure the Docker certificate before running the push command to access customer registry through HTTPS, otherwise, docker push command may fail.

5.1.2.2 Pushing the NEF Images to OCI Registry

NEF deployment package includes ready-to-use images and Helm charts to orchestrate containers in Kubernetes.

The following table lists the docker images for NEF:

Table 5-3 Docker Images for NEF

Service Name	Docker Image Name	Image Tag
Expiry Auditor Service	oc_nef_expiry_auditor	24.2.2
API Router	oc_nef_aef_apirouter	24.2.2

Table 5-3 (Cont.) Docker Images for NEF

Service Name	Docker Image Name	Image Tag
CCF Client	oc_nef_ccfclient_manager	24.2.2
Monitoring Events	oc_nef_monitoring_events	24.2.2
Quality of Service	oc_nef_quality_of_service	24.2.2
Traffic Influence	oc_nef_traffic_influence	24.2.2
Diameter Gateway	oc_nef_diam_gateway	24.2.2
Nrf Client Service	nrf-client	24.2.2
5GC Agent	oc_nef_5gcagent	24.2.2
APD Manager	oc_nef_apd_manager	24.2.2
Configuration Update Service	configurationupdate	24.2.15
Configuration INIT Service	configurationinit	24.2.15
Ingress Gateway	ocingress_gateway	24.2.15
Egress Gateway	ocegress_gateway	24.2.15
Common Config Hook	common_config_hook	24.2.15
Application Performance Service	oc-perf-info	24.2.2
Application Info Service	oc-app-info	24.2.2
Config-Server Service	oc-config-server	24.2.2
Debug Tools Service	ocdebug-tools	24.2.2
NF Test Service	nf_test	24.2.2
Device Trigger	oc_nef_device_trigger	24.2.2
Pool Manager	oc_nef_pool_manager	24.2.2
Camara	oc_camara_atf	24.2.2
Console Data Service	oc_nef_console_data_service	24.2.2
MSISDNLess MO SMS	oc_nef_msisdnless_mo_sms	24.2.2

Pushing Docker Images

Prerequisite: Download and untar the NEF Package ZIP file. For more information about downloading the package, see Installation Package Download.

To push the images to the registry:

1. Unzip the release package to the location where you want to install NEF. The package is as follows:
   `ocnef-pkg-24.2.2.0.0.tgz`
2. Untar the NEF package zip file to get NEF image tar file:

   ```
   tar -xvzf ocnef-pkg-24.2.2.0.0.tgz
   ```

   The directory consists of the following:

   - `ocnef-24.2.2.tgz`: Helm Chart
   - `ocnef-24.2.2.tgz.sha256`: Checksum for Helm chart tgz file
   - `ocnef-images-24.2.2.tar`: NEF Images File
   - `ocnef-images-24.2.2.tar.sha256`: Checksum for images tar file
   - `ocnef-network-policy-24.2.2.tgz`: Helm Chart for network policy

- `ocnef-network-policy-24.2.2.tgz.sha256` Checksum for network policy Helm chart tgz file

3. Run one of the following commands to load the `ocnef-images-24.2.2.tar` file:

```
docker load --input /IMAGE_PATH/ocnef-images-24.2.2.tar
```

```
podman load --input /IMAGE_PATH/ocnef-images-24.2.2.tar
```

4. Run one of the following commands to verify the images are loaded:

```
docker images
```

```
podman images
```

> **Note**
>
> Verify the list of images shown in the output with the list of images shown in the above table. If the list does not match, reload the image tar files.

5. Run one of the following commands to tag the images to the registry:

```
docker tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
```

```
podman tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>
```

6. Run one of the following commands to push the images to the registry:

```
docker push <docker-repo>/<image-name>:<image-tag>
```

```
podman push <docker-repo>/<image-name>:<image-tag>
```

> **Note**
>
> It is recommended to configure the Docker certificate before running the push command to access customer registry through HTTPS, otherwise, docker push command may fail.

5.1.2.3 Installing NEF Package

To install the NEF package, perform the following steps:

1. Run the following command to access the extracted package:

```
cd ocnef-<release_number>
```

For example:

```
cd ocnef-24.2.2.0.0
```

2. Customize the `ocnef-24.2.2-custom-values-ocnef.yaml` file with the required deployment parameters. See Customizing NEF chapter to customize the file. For more information about predeployment parameter configurations, see Preinstallation.
3. Run the following command to install NEF:

   ```
   helm install -f <custom-file> <release_name> <helm-chart> --namespace
   <release_namespace> --timeout 10m
   ```

   For example:

   ```
   helm install -f ocnef-24.2.2-custom-values-ocnef.yaml ocnef /home/cloud-
   user/ocnef-24.2.2.tgz --namespace ocnef
   ```

   where:

   - *helm_chart* is the location of the helm chart extracted from package file.
   - ocnef-24.2.2.tgz is the helm chart.
   - *release_name* is the release name used by helm command.

     Note

     – The *release_name* should not exceed 63 character limit.

     – In case of georedundant setup, it is mandatory to use unique *release_name* for each NEF instance.

   - *release_namespace* is the deployment namespace used by helm command.
   - *custom-file* is the name of the custom values yaml file (including location).

   Note

   - You can verify the installation while running the install command by entering this command on a separate terminal:

     ```
     watch kubectl get jobs,pods -n release_namespace
     ```

   - The DB hooks start creating the NEF database tables, once the `helm install` command is run.

   Following are the optional parameters that can be used in the `helm install` command:

   - **atomic:** If this parameter is set, installation process purges chart on failure. The `--wait` flag will be set automatically.
   - **wait:** If this parameter is set, installation process will wait until all pods, PVCs, Services, and minimum number of pods of a deployment, StatefulSet, or ReplicaSet

are in a ready state before marking the release as successful. It will wait for as long as `--timeout`.

- **timeout duration:** If not specified, default value will be 300 seconds in Helm. It specifies the time to wait for any individual Kubernetes operation (like Jobs for hooks). If the `helm install` command fails at any point to create a Kubernetes object, it will internally call the purge to delete after the timeout value. Here, the timeout value is not for overall installation, but for automatic purge on installation failure.

> ⚠ **Caution**
>
> Do not exit from `helm install` command manually. After running the `helm install` command, it takes some time to install all the services. In the meantime, you must not press **Ctrl+C** to come out from the command. It may lead to some anomalous behavior.

4. Press **Ctrl+C** to exit watch mode. Run the `watch` command on another terminal. Run the following command to check the status:

   ```
   helm status release_name -n release_namespace
   ```

5.1.3 Postinstallation Tasks

This section explains the postinstallation tasks for NEF.

5.1.3.1 Verifying NEF Installation

To verify the NEF installation:

1. Run the following command:

   ```
   helm status <helm-release> -n <namespace>
   ```

   Where,

   `<helm-release>` is the Helm release name of NEF.

   `<namespace>` is the namespace of NEF deployment.

   For example:

   ```
   helm status ocnef -n ocnef_namespace
   ```

   In the output, if `STATUS` is showing as `deployed`, then the installation is successful.

   Sample output:

   ```
   NAME: ocnef
   LAST DEPLOYED: Fri Sep 18 10:08:03 2020
   NAMESPACE: ocnef
   STATUS: deployed
    REVISION: 1
   ```

2. Run the following command to verify if the pods are up and active:

   ```
   kubectl get jobs,pods -n <Namespace>
   ```

   Where,

   <Namespace> is the namespace where NEF is deployed.

   For example:

   ```
   kubectl get pod -n ocnef
   ```

   In the output, the STATUS column of all the pods must be Running and the READY column of all the pods must be n/n, where n is the number of containers in the pod.

3. Run the following command to verify if the services are deployed and active:

   ```
   kubectl get services -n <Namespace>
   ```

   For example:

   ```
   kubectl get services -n ocnef_namespace
   ```

> **Note**
>
> If the installation is unsuccessful or the STATUS of all the pods is not in the Running state, perform the troubleshooting steps provided in the *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

5.1.3.2 Performing Helm Test

This section describes how to perform sanity check for NEF installation through Helm test. The pods to be checked should be based on the namespace and label selector configured for the Helm test configurations.

Helm Test is a feature that validates successful installation of NEF and determines if the NF is ready to take traffic.

> **Note**
>
> Helm Test expects all of the pods of given microservice to be in READY state for a successful result. However, the NRF Client Management microservice comes with Active/Standby mode for the multi-pod support in the current release. When the multi-pod support for NRF Client Management service is enabled, you may ignore if the Helm Test for NRF-Client-Management pod fails.

1. Complete the Helm test configurations under the "Helm Test Global Parameters" section of the oc-nef-24.2.2-custom-values.yaml file.

   ```
   nfName: ocnef
       image:
   ```

```
      name: nf_test
      tag: 24.2.2
      registry: cgbu-cnc-comsvc-release-docker.dockerhub-
phx.oci.oraclecorp.com/cgbu-ocudr-nftest
    config:
      logLevel: WARN
      timeout: 120      #Beyond this duration helm test will be considered
failure
    resources:
    - horizontalpodautoscalers/v1
    - deployments/v1
    - configmaps/v1
    - prometheusrules/v1
    - serviceaccounts/v1
    - poddisruptionbudgets/v1
    - roles/v1
    - statefulsets/v1
    - persistentvolumeclaims/v1
    - services/v1
    - rolebindings/v1
    complianceEnable: true
```

For more information on Helm test parameters, see Global Parameters.

2. Run the following command to perform the helm test:

```
helm test <release_name> -n <namespace>
```

where:

<release_name> is the release name.

namespace is the deployment namespace where NEF is installed.

For example:

```
helm test ocnef -n ocnef
```

Sample output:

```
NAME: ocnef
LAST DEPLOYED: Fri Nov 12 10:08:03 2020
NAMESPACE: ocnef
STATUS: deployed
REVISION: 1
TEST SUITE:     ocnef-test
Last Started:   Fri Nov 12 10:41:25 2020
Last Completed: Fri Nov 12 10:41:34 2020
Phase:          Succeeded
```

If the Helm test fails, see *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

5.1.3.3 Taking a Backup

Take a backup of the following files, which are required during fault recovery:

- Updated `oc-nef-24.2.2-custom-values.yaml` file
- Updated helm charts
- Secrets, certificates, and keys that are used during installation

5.3 Upgrading NEF

This section provides information about upgrading NEF deployment to the latest release.

Note

- In a georedundant deployment, perform the steps explained in this section on all the georedundant sites.
- Before starting with the NEF upgrade, ensure that CAPIF upgrade procedure is successfully complete. For information about the procedure to upgrade CAPIF, see Upgrade Tasks.
- To avoid any duplicate subscription for a Monitoring Event (ME) request, NEF validates the subscription request for any duplicate entries based on AFID, MSISDN, External ID, and MonitoringType parameters. NEF performs this validation only before and after the upgrade to NEF 24.2.2 and not during the upgrade process.
- If cnDBTier 24.2.2 is used during upgrade, set the `ndb_allow_copying_alter_table` parameter to `'ON'` in the `ocnef_dbtier_24.2.2_custom_values_24.2.2.yaml` file before upgrading NEF. After NEF upgrade, set the parameter to its default value `'OFF'`.
- Before upgrading, perform sanity check using Helm test. See Performing Helm Test section for the Helm test procedure.

5.3.1 Supported Upgrade Paths

The following table lists the supported upgrade paths for NEF.

Table 5-4 Supported Upgrade Paths

Source NEF release	Target NEF release
24.1.x	24.2.2
23.4.x	24.2.2

Note

NEF must be upgraded before upgrading cnDBTier.

5.3.2 Upgrade Tasks

This section provides information about the sequence of tasks to be performed for upgrading an existing NEF deployment.

> **Note**
>
> - It is recommended to perform NEF upgrade in a specific order. For more information about the upgrade order, see *Oracle Communications Cloud Native Core, Solution Upgrade Guide*.
> - Install or upgrade the network policies, if applicable. For more information, see Configuring Network Policies for NEF.

When you attempt to upgrade an existing NEF deployment, the running set of containers and pods are replaced with the new set of containers and pods. However, If there is no change in the pod configuration, the running set of containers and pods are not replaced.

If you need to change any configuration then change the oc-nef-24.2.2-custom-values.yaml file with new values.

> **Note**
>
> It is advisable to create a backup of the file before changing any configuration.

To configure the parameters, see Customizing NEF.

Run the following command to upgrade an existing NEF deployment:

```
helm upgrade <release> <chart> -f oc-nef-24.2.2-custom-values.yaml -f
OCCAPIF_API_Invoker_Mapping.yaml
```

> **Note**
>
> To avoid any duplicate subscription for a Monitoring Event (ME) request, NEF validates the subscription request for any duplicate entries based on AFID, MSISDN, External ID, and MonitoringType parameters. NEF performs this validation only before and after the upgrade to NEF and not during the upgrade process.

Table 5-5 Parameters and Definitions during NEF Upgrade

Parameters	Definitions
`<chart>`	It is the name of the chart that is of the form `<repository/ocnef>`. For example: `reg-1/ocnef or cne-repo/ocnef`
`<release>`	It can be found in the output of `helm list` command
`maxsurge`	
`maxUnavailability`	

In case of backout:

1. Check the history of helm deployment:
   `helm history <helm_release>`
2. Rollback to the required revision:
   `helm rollback <release name> <revision number>`

> **Note**
>
> Perform sanity check using Helm test. See Performing Helm Test section for the Helm test procedure.

5.4 Rolling Back NEF

This section provides information about rolling back NEF deployment to the previous release.

> **Note**
>
> - In a georedundant deployment, perform the steps explained in this section on all georedundant sites separately.
> - If operator chooses to rollback to a previous release or if some issue occurs during deployment and if the **Converged SCEF-NEF** feature is enabled, then the subscriptions on 4G side (HSS/MME) might not get cleaned up. To avoid this, operator should perform the following tasks:
>   - Before rollback, should trigger unsubscribe for all subscriptions created after upgrade
>   - After rollback, create new subscribe for the deleted subscriptions.
>   New subscription is not applicable for EPC network.
> - For upgrade or rollback, a default timeout of 5 minutes is provided by Helm. In case of a slow server, it might take more time. In such scenarios, the upgrade status will be `pending-upgrade` or `pending-rollback`.
>   To avoid this, add `--timeout` flag according to their network speed.
>
>   **Example**
>
>   Upgrade:
>
>   ```
>   helm upgrade oncef oc-nef -f experiment.yaml –-timeout=45m -n mv --
>   dry-run
>   ```
>
>   Rollback:
>
>   ```
>   helm rollback oncef 1 -n mv –-timeout=45m
>   ```
>
> - To avoid any duplicate subscription for a Monitoring Event (ME) request, the NEF validates subscription request for any duplicate entries based on AFID, MSISDN, External ID and MonitoringType parameters. NEF does not perform this validation if NEF is rolled back from 24.2.2 to any previous release. The validation resumes when NEF is upgraded back to 24.2.2.

5.4.1 Supported Rollback Paths

The following table lists the supported rollback paths for NEF:

Table 5-6 Supported Rollback Paths

Source Release	Target Release
24.2.2	24.1.x
24.2.2	23.4.x

> **Note**
>
> If georedundancy feature was disabled before upgrading to 23.2.x, then rolling back to a previous version will automatically disable this feature. However, the database will still have records of the `NfInstances` and `NfSubscriptions` from the mated sites. For more information, contact My Oracle Support.

5.4.2 Rollback Tasks

To roll back from NEF 24.1.x to a previous version:

1. Run the following command to check the revision you must roll back to:

   ```
   $ helm history <release_name> -n <release_namespace>
   ```

2. Run the command to rollback to the required revision:

   ```
   $ helm rollback <release_name> <revision_number> -n <release_namespace>
   ```

> **Note**
>
> - No configuration should be performed during rollback.
> - Do not exit from `helm rollback` command manually. After running the `helm rollback` command, it takes some time (depending upon number of pods to rollback) to rollback all of the services. In the meantime, you must not press "**ctrl+c**" to come out from `helm rollback` command. It may lead to anomalous behavior.
> - Ensure that no NEF pod is in the failed state.
> - If the rollback is not successful, perform the troubleshooting steps mentioned in *Oracle Communications Cloud Native Core, Network Exposure Function Troubleshooting Guide*.

5.2 Customizing NEF

This section provides information about customizing Oracle Communications Cloud Native Core, Network Exposure Function (NEF) deployment in a cloud native environment.

The NEF deployment is customized by overriding the default values of various configurable parameters in the oc-nef-24.2.2-custom-values.yaml file.

Basic Configurations

- Once Docker platform configurations are done, proceed as per Configurable Parameters of NEF.
- Check Registry is in place and contains latest Helm charts and jar as per the release for NEF node.

Perform the following steps to customize the oc-nef-24.2.2-custom-values.yaml file as per the required parameters:

1. Unzip the `Custom_Templates` file available in the extracted documentation release package to get the following files that are used to customize the deployment parameters during installation:
   - `oc-nef-24.2.2-custom-values.yaml`: This file is used to customize the deployment parameters during installation.
   - 16 custom value files for NEF deployment with ATS: NEF provides sixteen custom value files that can be used for customizing the deployment parameters when NEF is deployed with Automated Testing Suite (ATS). Each file allows you to run NEF testcases on ATS based on the mapping between the supported communication models and NEF feature configurations.

     NEF provides the following files to be used for ATS:
     - `oc-nef-24.2.2-custom-values.yaml`
     - `oc-nef-24.2.2-custom-values-ats.yaml`
     - oc-nef-custom-values-ats-bsf-modelA
     - oc-nef-custom-values-ats-bsf-modelB
     - oc-nef-custom-values-ats-bsf-modelD
     - oc-nef-custom-values-ats-pcf-gmlcfailoverudm-modelA.yaml
     - oc-nef-custom-values-ats-pcf-gmlcfailoverudm-modelB.yaml
     - oc-nef-custom-values-ats-pcf-gmlcfailoverudm-modelD.yaml
     - oc-nef-custom-values-ats-udm-udr-modelA.yaml
     - oc-nef-custom-values-ats-udm-udr-modelB.yaml
     - oc-nef-custom-values-ats-udm-udr-modelD.yaml
     - ocnef-network-policy-custom-values

     For more information about NEF ATS, see *Oracle Communications Cloud Native Core, Automated Testing Suite Guide*.

     > **Note**
     >
     > Use this file for customization, if NEF is being deployed for running ATS test cases in the same namespace as ATS deployment.

   - `NefAlertrules-24.2.2.yaml`: This file is used for `Prometheus`.
   - `NefDashboard-24.2.2.json`: This file is used by `Grafana`.

> **Note**
>
> For information about the metrics added, see *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

- `ocnef_oci_alertrules_24.2.2.zip`: This file is used for creating alerts from OCI terraform files.
- `OCCAPIF_API_Invoker_Mapping.yaml`: This file is used to perform the preupgrade tasks during NEF upgrade. For more information about using this file, see Preinstallation.
- `ocnef_dbtier_24.2.2_custom_values_24.2.2.yaml`: This file is used to install NEF with the recommended resource requirements for cnDBTier.

For more information on how to download the package from My Oracle Support, see Installation Package Download section.

2. Customize the oc-nef-24.2.2-custom-values.yaml file.
3. Save the updated oc-nef-24.2.2-custom-values.yaml file in the Helm chart directory.

> **Note**
>
> - All parameters mentioned as mandatory must be present in Custom Values YAML file.
> - All fixed value parameters listed must be present in the Custom Values YAML file with the exact values as specified in this section.

For more information on the configurable parameters, see Configurable Parameters of NEF.

5.2.1 Configurable Parameters of NEF

This section includes information about the configuration parameters of NEF.

NEF allows customization of parameters for the different microservices and related settings.

> **Note**
>
> - Mandatory parameters must be configured before the NEF deployment.
> - During installation, all the configurations would be read from Helm. Any configurations that support **Update** operation through REST API can only be updated using REST API or Console and any further updates to these configurations using Helm would be ignored. For further information, refer to *Oracle Communications Cloud Native Core, Network Exposure Function REST Specification Guide* and *Configuring Network Exposure Function using the CNC Console* chapter in *Oracle Communications Cloud Native Core, Network Exposure Function User Guide*.

5.2.1.1 Global Parameters

Table 5-7 Global Parameters

Parameter	Description	Details
`dockerRegistry`	Specifies the name of the Docker registry, which hosts CNC Policy docker images.	This is a docker registry running CNE bastion server where all OAuth docker images are loaded. This is a mandatory parameter.
`vendor`	The vendor name.	This is a mandatory parameter. **Default Value:** Oracle
`serviceAccountName`	Name of the service account for NEF.	This is an optional parameter. **Note:** The value for the `serviceAccountName` parameter must be same as the service account name configured for NEF. For more information, see Creating Service Account, Role, and RoleBinding.
`app_name`	Name of the application.	This is a mandatory parameter. **Default Value:** ocnef
`nefK8NameSpace`	The Kubernetes namespace.	
`prometheusUrl`	The prometheus URL.	
`nfInstanceId`		
`nefSiteId`		
`siteOffSetId`	It should be a unique sequential ID for each site in a GR deployment. Each site should be assigned with the unique auto-increment offsetvalue. This is a mandatory parameter.	For example, in a 4 site deployment, each site's `siteOffSetId` value should be assigned within the range from 1 to 4.
`nefExtIngressEndPoint`	Provide the IP Endpoint or FQDN of the NF.	**Default Value:** Service Name of Ext IGW if kept empty **Example:** ext.ingress.ocnef.5gc.mnc014.mcc310.3gppnetwork.org

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
nefFivegcIngressEndPoint	Provide the IP Endpoint or FQDN of the NF.	**Default Value:** Service Name of 5gc IGW if kept empty **Example:** fivegc.ingress.ocnef.5gc.mnc014.mcc310.3gppnetwork.org
serviceMeshCheck	Specifies whether to enable or disable Service Mesh Check.	Set this flag to true when service required else false. **Default Value:** false
nefApiPrefix	This is the prefix set for all NEF N33 & T8 APIs.	This is a mandatory parameter. **Default Value:** ' ' **Note:** This is not applicable for notification.
scefDiamHost	The parameter provides the IP Endpoint or FQDN of the Diameter Gateway.	**Default Value:** Service Name of DiameterGateway if kept empty
envDiameterRealm	Diameter Realm of Gateway provided in CE messages.	**Default Value:** "statefulset name" if kept empty
envDiameterIdentity	Diameter Identity of Gateway provided in CE messages.	**Default Value:** "pod name in statefulset" if kept empty
enableFeature.convergedScefNef	This parameter enables or disables Converged SCEF NEF service.	Enabling this flag introduces Converged_scef_nef diameter gateway for receiving diameter traffic from EPC network nodes. Enables EPC subscription parameters for subscriptions towards UDM. **Default Value:** false
enableFeature.gmlc	Specifies whether to enable or disable GMLC as the location provider.	Set this flag to true, if GMLC is deployed within the same network as NEF. **Default Value:** false

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
enableFeature.trafficInfluence	This parameter enables or disables Traffic Influence service.	Set this flag to false when Traffic Influence service is not required. **Default Value:** true
enableFeature.deviceTrigger	Set this flag to true when Device Trigger service is required.	**Default Value:** false
enableFeature.msisdnless_mo_sms	Set this flag to true when MSISDNless MO service is required.	**Default Value:** false
commonCfg.hookImage.name	Common Config hook image name.	**Default Value:** common_config_hook
commonCfg.hookImage.tag	Common Config hook image tag version.	This is a mandatory parameter.
commonCfg.hookImage.pullPolicy	Common Config hook image pull policy status.	**Default Value:** Always
client.enable	This enables the hook jobs to register svc to CDS and to enable to config client in service.	**Default Value:** true
client.pollingInterval	Specifies interval at which config client query the config server for configuration.	**Default Value:** 1000
server.svcName	This configuration to be used by GW. NEF derives this host and port dynamically.	This is a mandatory parameter.
server.port	Common Config server port number.	**Default Value:** 8080

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`capifDetailsMonitorDelay`	Specifies delay between successive invocations to read the updated secret for configured CAPIF's.	**Default value:** 15000
`publicKeyMonitorDelay`	Specifies delay between successive invocations to read updated CAPIF details from API Router Config Map.	**Default value:** 15000
`capifDetails.capifInstanceId`	This parameter identifies the available CAPIF instances in a georedundant deployment.	
`capifDetails.type`	Specifies if the CAPIF instance is local or from a georedundant setup.	**Default value:** local
`capifDetails.host`	The host details for the CAPIF.	**Default value:** occapif-network-ingressgateway
`capifDetails.port`	The port details for the CAPIF.	**Default value:** 80
`capifDetails.schema`	Specifies if the schema is http or https.	**Default value:** http

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`capifDetails.apiPrefix`	When CAPIF has an API prefix, this parameter allows NEF to communicate with CAPIF using the prefix added to the path. **Note:** While upgrading, ensure it matches the corresponding CAPIF's `capifApiPrefix` global configuration parameter.	**Default value:** ' ' **Note:** While upgrading to 23.3.x from 23.1.x or above release, ensure the default value is updated from " " to "apiRoot". This is required to match the default value of older release(s).
`capifDetails.certificate.secretName`	The certificate details of the CAPIF.	**Default value:** certificate-secret
`capifDetails.certificate.certificateName`	Name of the CAPIF certificate.	**Default value:** tmp.cer
`jaegerTelemetryTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing	**Default Value:** false
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5
`openTelemetry.jaeger.httpExporter.host`	Specifies the host of Jaeger collector service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger collector service	**Default Value:** 4318
mTLS Configurations		

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
externalGWConfig.initSSL	Enable this to enable mtls on external gateways that includes both Ingress GW (incoming) and Egress GW (outgoing) of NEF EG.	This value must be set to **true** only if the value for the externalGWConfig.igw.enableIncomingHttp parameter is **false**. **Default Value:** true **Note:** For NEF, the mtls cannot be enabled for both external gateways and 5GC gateways simeltaneously. Therefore, only one of the parameters between externalGWConfig.initSSL and fivegcGWConfig.initSSL can be set to **true**.
externalGWConfig.igw.enableIncomingHttp	Enables HTTP requests on northbound side (extEnableIncomingHttp)	**Default Value:** false **Note:** Either externalGWConfig.initSSL or externalGWConfig.igw.enableIncomingHttp must be set to **true**.
externalGWConfig.igw.publicHttpSignallingPort		**Default Value:** 80
externalGWConfig.igw.publicHttpsSignallingPort		**Default Value:** 443
externalGWConfig.egw.publicHttpsSignallingPort		**Default Value:** 8080
externalGWConfig.tls.privateKey.k8SecretName	Name of the Kubernetes secret object containing ext-expgw-secret username and password	This is an optional parameter. **Default Value:** ext-expgw-secret
externalGWConfig.tls.privateKey.k8NameSpace	The namespace of the Kubernetes secret object containing ext-expgw-secret.	This is an optional parameter.
externalGWConfig.tls.privateKey.rsa.filename	The filename containing rsa keydetail of the ext-expgw-secret.	This is an optional parameter. **Default Value:** rsa_private_key_pkcs1.pem

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`externalGWConfig.tls.privateKey.ecdsa.filename`	The filename containing ecdsa keydetail of the ext-expgw-secret.	This is an optional parameter. **Default Value:** ssl_ecdsa_private_key.pem
`externalGWConfig.tls.certificate.k8SecretName`	Name of the Kubernetes secret object containing ext-expgw-secret certificate.	This is an optional parameter. **Default Value:** ext-expgw-secret
`externalGWConfig.tls.certificate.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-expgw-secret certificate.	This is an optional parameter.
`externalGWConfig.tls.certificate.rsa.filename`	The filename containing rsa keydetail of the ext-expgw-secret certificate.	This is an optional parameter. **Default Value:** tmp.cer **Note:** Use the same certificate name for the AEF API Router certificate name in the . For more information about the AEF API Router configurations, see AEF API Router Parameters.
`externalGWConfig.tls.certificate.ecdsa.filename`	The filename containing ecdsa keydetail of the ext-expgw-secret certificate.	This is an optional parameter. **Default Value:** ssl_ecdsa_certificate.crt
`externalGWConfig.tls.caBundle.k8SecretName`	Name of the Kubernetes secret object containing ext-exp-gw CA details for truststore.	This is an optional parameter. **Default Value:** ext-expgw-secret
`externalGWConfig.tls.caBundle.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-exp-gw CA details for truststore.	This is an optional parameter.

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
externalGWConfig.tls.caBundle.filename	The filename containing ext-exp-gw CA details for truststore.	This is an optional parameter. **Default Value:** caroot.cer
externalGWConfig.tls.keyStorePassword.k8SecretName	Name of the Kubernetes secret object containing ext-exp-gw KeyStore password	This is an optional parameter. **Default Value:** ext-expgw-secret
externalGWConfig.tls.keyStorePassword.k8NameSpace	The namespace of the Kubernetes secret object containing ext-exp-gw KeyStore password	This is an optional parameter.
externalGWConfig.tls.keyStorePassword.filename	The filename containing ext-exp-gw KeyStore password	This is an optional parameter. **Default Value:** key.txt
externalGWConfig.tls.trustStorePassword.k8SecretName	Name of the Kubernetes secret object containing ext-exp-gw truststore password	This is an optional parameter. **Default Value:** ext-expgw-secret
externalGWConfig.tls.trustStorePassword.k8NameSpace	The namespace of the Kubernetes secret object containing ext-exp-gw truststore password	This is an optional parameter.
externalGWConfig.tls.trustStorePassword.filename	The filename containing ext-exp-gw truststore password	This is an optional parameter. **Default Value:** trust.txt
externalGWConfig.tls.initialAlgorithm	The initial algorithm selected by ext-exp-gw.	Possible values are: • ES256 • RS256 **Default Value:** RS256

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`externalGWConfig.tls.keyType`	The selected key type.	Possible values are: • rsakey • ecdsakey **Default Value:** rsakey
5GC GW Configurations		
`fivegcGWConfig.initSSL`	Enable this to enable mTLS on 5GC gateways that includes both Ingress GW (incoming) and Egress GW (outgoing) of NEF.	This value must be set to **true** only if the value for the `fivegcGWConfig.igw.enableIncomingHttp` parameter is **false**. **Default Value:** true **Note:** For NEF, the mtls cannot be enabled for both external gateways and 5GC gateways simeltaneously. Therefore, only one of the parameters between `externalGWConfig.initSSL` and `fivegcGWConfig.initSSL` can be set to **true**.
`fivegcGWConfig.igw.enableIncomingHttp`	Enables HTTP requests on northbound side (extEnableIncomingHttp)	When this is enabled, it is recommended to disable the `externalGWConfig.igw.enableIncomingHttp` parameter as only HTTPS should be supported. **Default Value:** false **Note:** Either `fivegcGWConfig.initSSL` or `fivegcGWConfig.igw.enableIncomingHttp` must be set to **true**.
`fivegcGWConfig.igw.publicHttpSignallingPort`		**Default Value:** 80
`fivegcGWConfig.igw.publicHttpsSignallingPort`		**Default Value:** 443
`fivegcGWConfig.egw.publicHttpsSignallingPort`		**Default Value:** 8080
`fivegcGWConfig.tls.privateKey.k8SecretName`	Name of the Kubernetes secret object containing fivegc-service-secret username and password	This is an optional parameter. **Default Value:** fivegc-service-secret

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`fivegcGWConfig.tls.privateKey.k8NameSpace`	The namespace of the Kubernetes secret object containing ext-fivegc-service-secret.	This is an optional parameter.
`fivegcGWConfig.tls.privateKey.rsa.filename`	The filename containing rsa keydetail of the fivegc-service-secret.	This is an optional parameter. **Default Value:** rsa_private_key_pkcs1.pem
`fivegcGWConfig.tls.privateKey.ecdsa.filename`	The filename containing ecdsa keydetail of the expgw-secret.	This is an optional parameter. **Default Value:** ssl_ecdsa_private_key.pem
`fivegcGWConfig.tls.certificate.k8SecretName`	Name of the Kubernetes secret object containing expgw-secret certificate.	This is an optional parameter. **Default Value:** fivegc-service-secret
`fivegcGWConfig.tls.certificate.k8NameSpace`	The namespace of the Kubernetes secret object containing fivegc-service-secret certificate.	This is an optional parameter.
`fivegcGWConfig.tls.certificate.rsa.filename`	The filename containing rsa keydetail of the fivegc-service-secret certificate.	This is an optional parameter. **Default Value:** tmp.cer
`fivegcGWConfig.tls.certificate.ecdsa.filename`	The filename containing ecdsa keydetail of the fivegc-service-secret certificate.	This is an optional parameter. **Default Value:** ssl_ecdsa_certificate.crt

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`fivegcGWConfig.tls.caBundle.k8SecretName`	Name of the Kubernetes secret object containing fivegc-service-secret CA details for truststore.	This is an optional parameter. **Default Value:** fivegc-service-secret
`fivegcGWConfig.tls.caBundle.k8NameSpace`	The namespace of the Kubernetes secret object containing fivegc-service-secret CA details for truststore.	This is an optional parameter.
`fivegcGWConfig.tls.caBundle.filename`	The filename containing fivegc-service-secret CA details for truststore.	This is an optional parameter. **Default Value:** caroot.cer
`fivegcGWConfig.tls.keyStorePassword.k8SecretName`	Name of the Kubernetes secret object containing fivegc-service-secret KeyStore password	This is an optional parameter. **Default Value:** fivegc-service-secret
`fivegcGWConfig.tls.keyStorePassword.k8NameSpace`	The namespace of the Kubernetes secret object containing fivegc-service-secret KeyStore password	This is an optional parameter.
`fivegcGWConfig.tls.keyStorePassword.filename`	The filename containing fivegc-service-secret KeyStore password	This is an optional parameter. **Default Value:** fivegc-service-secret

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`fivegcGWConfig.tls.trustStorePassword.k8SecretName`	Name of the Kubernetes secret object containing fivegc-service-secret truststore password	This is an optional parameter.
`fivegcGWConfig.tls.trustStorePassword.k8NameSpace`	The namespace of the Kubernetes secret object containing fivegc-service-secret truststore password	This is an optional parameter. **Default Value:**&trustStorePasswdSecretNameSpace default
`fivegcGWConfig.tls.trustStorePassword.filename`	The filename containing fivegc-service-secret truststore password	This is an optional parameter. **Default Value:**&trustStorePasswdFileName trust.txt
`fivegcGWConfig.tls.initialAlgorithm`	The initial algorithm selected by fivegc-service-secret.	Possible values are: • ES256 • RS256 **Default Value:** RS256
`fivegcGWConfig.tls.keyType`	The selected key type.	Possible values are: • rsakey • ecdsakey **Default Value:** rsakey
Database Configurations		
`database.dbName`	Name of the service database	This is a mandatory parameter. **Note:** The parameter value must be same for all the NEF instances in a georedundant depolyment.
`database.releaseDbName`	Name of the release database containing release version details	**Note:** The parameter value must be unique for all the NEF instances in a georedundant depolyment.
`database.nrfClientDbName`	It is a NRF Client specific DB. It should be unique across the GR deployment.	**Note:** When upgrading from earlier releases, for this parameter ensure to use the same name used in the respective release's `database.dbName`.

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`database.dbPrimaryHost`	The primary host details for the database	This is a mandatory parameter.
`database.dbSecondaryHost`	The secondary host details for the database	This is a mandatory parameter.
`database.dbPort`	Database Port details	
`database.appUserSecretName`	Name of the Kubernetes secret object containing Database username and password for an application user	This is a mandatory parameter. **Default Value:** appuser-secret
`database.privilegedUserSecretName`	Name of the Kubernetes secret object containing Database username and password for an privileged user	This is a mandatory parameter. **Default Value:** privilege-user-secret
`database.dropSchema`	Drop Schema during helm delete for all services	false
`database.dbUNameLiteral`	Name of the key configured for "DB Username" in appuser-secret.	This is a mandatory parameter.
`database.dbPwdLiteral`	Name of the key configured for "DB Password" in appuser-secret.	This is a mandatory parameter.

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`database.engine`	The database engine.	Any value other than InnoDB or NDBCluster leads to failure of database table creation process. This is a mandatory parameter. **Default Value:** NDBCluster
`extraContainers`	The flag to enable or disable injecting extra container.	This is an optional parameter. **Default Value:** DISABLED
`debugToolContainerMemoryLimit`	Indicates the memory assigned for the debug tool container.	
NRF Client Configurations **Note:** These configurations are required when NEF is required to register with NRF. Before configuring NRF client configuration, you must enable NRF Client services.		
`nrfClientNfDiscoveryEnable`	Specifies whether to enable or disable NF Discovery service. The value for this parameter must be set to true if on demand discovery is required.	This is a mandatory parameter. **Default Value:** true
`nrfClientNfManagementEnable`	Specifies whether to enable or disable NF Management service.	This is a mandatory parameter. **Default Value:** true
`performanceServiceEnable`	Specifies whether to enable or disable performance service.	This is a mandatory parameter. **Default Value:** true
`appinfoServiceEnable`	Specifies whether to enable or disable the app info service.	This is a mandatory parameter. **Default Value:** true

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`configServerEnable`	Specifies whether to enable or disable the config-server service.	This is a mandatory parameter. **Default Value:** true
`envJaegerAgentHost`	Specifies the hostname or IP address for the jaeger agent	
`envJaegerAgentPort`	Specifies the port of Jaeger Agent service	
`envMysqlHost`	Specifies the IP address or host name of the MySql server which hosts NEF databases	
`envMysqlPort`	Port of the MySql server which hosts Cloud Native Core Policy's databases	
`deploymentNrfClientService.envNfNamespace`	Specifies the Kubernetes namespace of NEF	
`deploymentNrfClientService.envConsumeSvcName`	Specifies the NEF consumer service name for NRF Client service deployment	**Syntax:** *<Monitoring Event k8s service name>:3GPP defined service name'* **Sample value:** `<Helm Release Name>-monitoringevents:3gpp-monitoring-event'`
`deploymentNrfClientService.nfApiRoot`	API root of NEF. It is the Load Balancer IP or FQDN of 5G Ingress Gateway.	
Config-Server Configurations		

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`privilegedDbCredSecretName`	Name of the Kubernetes secret object containing Database username and password for an admin user	
`dbCredSecretName`	Name of the Kubernetes secret object containing Database username and password	
`releaseDbName`	Name of the release database containing release version details	
Prometheus Scraping Configuration		
`prometheusScrapingConfig.enabled`	Flag to enable or disable the Prometheus Scraping Configuration.	This is a mandatory parameter. **Default Value:** true
`prometheusScrapingConfig.path`	Prometheus scrap path	This is a mandatory parameter. **Default Value:** "/actuator/prometheus"
`type`	The type of service.	Possible values are: • ClusterIP • NodePort • LoadBalancer • ExternalName This is an optional parameter. **Default Value:** LoadBalancer
`customExtension.allResources.labels`	Custom Labels that needs to be added to all the NEF Kubernetes resources	This can be used to add custom label(s) to all Kubernetes resources that are created by NEF helm chart. This is an optional parameter.

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
customExtension.allResources.annotations	Custom Annotations that needs to be added to all the NEF Kubernetes resources	This can be used to add custom annotation(s) to all Kubernetes resources that are created by NEF helm chart. This is an optional parameter.
customExtension.lbServices.labels	Custom Labels that needs to be added to NEF services that are considered as Load Balancer type	This can be used to add custom label(s) to all Load Balancer Type Services that are created by NEF helm chart. This is an optional parameter.
customExtension.lbServices.annotations	Custom Annotations that needs to be added to NEF services that are considered as Load Balancer type	This can be used to add custom annotation(s) to all Load Balancer Type Services that are created by NEF helm chart. This is an optional parameter.
customExtension.lbDeployments.labels	Custom Labels that needs to be added to NEF deployments that are associated to a service which is of Load Balancer type	This can be used to add the custom label(s) to the deployments that will be created by NEF helm chart that are associated to a Load Balancer Type Service. This is an optional parameter.
customExtension.lbDeployments.annotations	Custom Annotations that needs to be added to NEF deployments that are associated to a service which is of Load Balancer type	This can be used to add the custom label(s) to the deployments that will be created by NEF helm chart that are associated to a Load Balancer Type Service. This is an optional parameter.

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
customExtension.nonlbServices.labels	Custom Labels that needs to be added to NEF Services that are considered as not Load Balancer type	This can be used to add custom label(s) to all non-Load Balancer Type Services that is created by NEF helm chart. This is an optional parameter.
customExtension.nonlbServices.annotations	Custom Annotations that needs to be added to NEF Services that are considered as not Load Balancer type	This can be used to add custom annotation(s) to all non-Load Balancer Type Services that is created by NEF helm chart. This is an optional parameter.
customExtension.nonlbDeployments.labels	Custom Labels that needs to be added to NEF Deployments that are associated to a Service which is not of Load Balancer type	This can be used to add custom label(s) to all Deployments that is created by NEF helm chart which are associated to a Service which if not of Load Balancer Type. This is an optional parameter.
customExtension.nonlbDeployments.annotations	Custom Annotations that needs to be added to NEF Deployments that are associated to a Service which is not of Load Balancer type	This can be used to add custom annotation(s) to all deployments that is created by NEF helm chart that are associated to a service which is not of Load Balancer Type. This is an optional parameter.
Helm Test Hook Configurations		
test.nfName	Name of the NF	This is a mandatory parameter. **Default Value:** ocnef
test.image.name	Image name for the test container	This is a mandatory parameter. **Default Value:** nf_test
test.image.tag	Tag for the test container image	This is a mandatory parameter.
test.config.logLevel	Set the logging level	This is a mandatory parameter. **Default Value:** WARN

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`test.config.timeout`	Specify timeout until test container checks for Pod's health	This is a mandatory parameter. **Default Value:** 20

`test.resources`	Specifies the helm test resources.	**Example:**
		`horizontalpodautoscalers/v1`
		`deployments/v1`
		`configmaps/v1`
		`prometheusrules/v1`
		`serviceaccounts/v1`
		`poddisruptionbudgets/v1`
		`roles/v1`
		`statefulsets/v1`
		`persistentvolumeclaims/v1`
		`services/v1`
		`rolebindings/v1`

`test.complianceEnable`	Enables or disables the helm test logging feature.	This is a mandatory parameter. Possible values are: • true • false **Default value:** false
`configurableErrorCodes.enabled`	Specifies if the configurable error codes must be enabled.	This is a mandatory parameter. Possible values are: • true • false **Default value:** false

`configurableErrorCodes.errorScenarios`	Contains a list of exceptionType, error codes, errorDescription, error causes, and errorTitle for which a failover must be performed.	**Example:**
		`configurableErrorCodes:`
		`enabled: false`
		`errorScenarios:`
		`exceptionType:`
		`"ConnectException"`
		`errorCode: "503"`
		`errorDescription:`
		`"Connection failure"`
		`errorCause: "Connection Refused"`
		`errorTitle:`
		`"ConnectException"`
		
		Note: The values in the `errorScenarios` parameter must be same as the values provided for the `monitoringevents.gmlc.switchOnErrorCodes` parameter.

Georedundancy Parameters		

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`geoRedundancyOptions.featureStatus`	Specifies if georedundancy feature must be enabled or not.	The value must be set to **DISABLED** in case of single site. **Note:** All the parameters under **geoRedundancyOptions** are applicable only when the value of this parameter is "ENABLED".
`geoRedundancyOptions.peerGRSiteList`	List of the georedundant sites with site name and site instance Id.	
`geoRedundancyOptions.handleNotification.checkSiteStatus`	Specifies if NEF must validate the status of the owner site before processing notification in non-owner site.	If the value is set to **false**, then the notifications are processed without validating the status of owner site.
`geoRedundancyOptions.handleNotification.checkDBReplicationStatus`	Specifies if NEF must validate the replication status before processing notification in non-owner site.	If the value is set to **false**, then notifications are processed without verifying the DB replication status.
Network Configuration Parameters		
`networkConfiguration.qosProfilesMap.<qosReference attribute value>.mediaType`	Specifies the `mediaType` value received in the AF request. `mediaType` defines the QoS reference to set the media type value.	Possible values are: • AUDIO • VIDEO • DATA • APPLICATION • TEXT • MESSAGE • OTHER **Note:** In case of matching `qosReference` is not found, `mediaType` is set to `OTHER` by default.
`networkConfiguration.qosProfilesMap.<qosReference attribute value>.maxReqBwDl`	It is the max requested bandwidth for downlink.	In max requested bandwidth for DL configuration, the format should be as follows: `^\d+(\.\d+)? (bps|Kbps|Mbps|Gbps|Tbps)$` For example: "1 Gbps", "100 Mbps"

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
networkConfiguration.qosProfilesMap.<qosReference attribute value>.maxReqBwUl	It is the max requested bandwidth for uplink.	In max requested bandwidth for UL configuration, the format should be as follows: ^\d+(\.\d+)? (bps|Kbps|Mbps|Gbps|Tbps)$ For example: "1 Gbps", "100 Mbps"
networkConfiguration.qosProfilesMap.<qosReference attribute value>.minReqBwDl	It is the min requested bandwidth for downlink.	In min requested bandwidth for DL configuration, the format should be as follows: ^\d+(\.\d+)? (bps|Kbps|Mbps|Gbps|Tbps)$ For example: "1 Gbps", "100 Mbps"
networkConfiguration.qosProfilesMap.<qosReference attribute value>.minReqBwUl	It is the min requested bandwidth for uplink.	In min requested bandwidth for UL configuration, the format should be as follows: ^\d+(\.\d+)? (bps|Kbps|Mbps|Gbps|Tbps)$ For example: "1 Gbps", "100 Mbps"
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>	Identifies a valid geographic zone that the AF request applies only to the traffic of UE(s) located in this specific zone.	Any number of zoneIds shall be configured aligning to the geoZoneIdToSpatialValidityMap section.
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.praId	Represents an identifier of the Presence Reporting Area.	
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.additionalPraId	This may be present if the praId IE is present and if it contains a PRA identifier referring to a set of Core Network predefined Presence Reporting Areas.	

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.presenceState	Indicates whether the UE is inside or outside of the area of interest	possibly anyone of values IN_AREA, OUT_OF_AREA, UNKNOWN, INACTIVE
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.trackingAreaList	Represents the list of tracking areas that constitutes the area and the values referenced here should be configured under section 'taiMap'	Minimum one to be configured
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.ecgiList	Represents the list of EUTRAN cell Ids that constitutes the area and the values referenced here should be configured under section 'ecgiMap'	Minimum one to be configured
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.ncgiList	Represents the list of NR cell Ids that constitutes the area and the values referenced here should be configured under section 'ncgiMap'	Minimum one to be configured
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.globalRanNodeIdList	Represents the list of NG RAN node identifiers that constitutes the area and the values referenced here should be configured under section 'globalRanNodeIdMap'	Minimum one to be configured

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
networkConfiguration.geoZoneIdToSpatialValidityMap.<validGeoZoneId>.globaleNbIdList	Represents the list of eNodeB identifiers that constitutes the area and the values referenced here should be configured under section 'globalRanNodeIdMap'	
networkConfiguration.plmnIdMap.<plmnId>	Represents the identity of the PLMN and corresponding 'mcc' and 'mnc' should be configured under it	
networkConfiguration.taiMap.<taiId>	Represents tracking area identity. plmnId and tac are mandate to be configured under this	
networkConfiguration.ecgiMap.<ecgiId>	Represents the EUTRAN cell Id that constitutes an area. plmnId and eutraCellId are mandate to be configured under this	
networkConfiguration.ncgiMap.<ncgiId>	Represents NR cell Id that constitutes the area. plmnId and nrCellId are mandate to be configured under this	

Table 5-7 (Cont.) Global Parameters

Parameter	Description	Details
`networkConfiguration.globalRanNodeIdMap.<RANnodeId>`	Represents a global RAN. One of n3IwfId, gNbId, ngeNbId, wagfId, tngfId, eNbId is mandate to be configured under this.	
`networkConfiguration.afServiceId.<serviceId>`	Represent the afservice identifier from AF and corresponding dnn, snnsai, routeToLoc configuration map. This configuration is applicable when DNN, SNSSAI config is absent from AF.	

5.2.1.2 Pool Manager Parameters

Table 5-8 Pool Manager Parameters

Parameter	Description	Notes
`image.name`	Pool Manager image name	This is an optional parameter. **Default Value:** oc_nef_pool_manager
`image.tag`	Pool Manager image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** Always
`jetty.serviceMeshCheck`	Load balancing will be handled by Ingress gateway, if true it would be handled by serviceMesh	
`jetty.connectionTimeout`	Jetty client connection timeout settings	**Default Value:** 10000
`jetty.requestTimeout`	Jetty client request timeout settings	**Default Value:** 12000

Table 5-8 (Cont.) Pool Manager Parameters

Parameter	Description	Notes
extraContainers		**Default Value:** USE_GLOBAL_VALUE
log.level.root	Log level for root logs	This is an optional parameter. **Default Value:** WARN
log.level.nef	Log level for NEF service logs	**Default Value:** INFO
resources.limits.cpu	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
resources.limits.hooksCpu	Maximum amount of hooks CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
resources.limits.memory	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi
resources.limits.hooksMemory	Maximum amount of hooks memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi
resources.requests.cpu	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
resources.requests.hooksCpu	The amount of hooks CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
resources.requests.memory	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
resources.requests.hooksMemory	The amount of hooks memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi
resources.target.averageCpuUtil		
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 12
maxUnavailable		This is an optional parameter. **Default Value:** 0

5.2.1.3 Monitoring Events Parameters

Table 5-9 Monitoring Events Parameters

Parameter	Description	Notes
`image.name`	monitoringevents image name	This is an optional parameter. **Default Value:** oc_nef_monitoring_events
`image.tag`	monitoringevents image tag	This is an optional parameter. **Default Value:** NEF Images
`jetty.requestTimeout`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2000ms
`loglevel.root`	Log level for root logs	**Default Value:** WARN
`loglevel.nef`	Log level for NEF service logs	**Default Value:** WARN
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
`resources.limits.hooksCpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi
`resources.limits.hooksMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
`resources.requests.hooksCpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
`resources.requests.hooksMemory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 60
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 12

Table 5-9 (Cont.) Monitoring Events Parameters

Parameter	Description	Notes
`maxUnavailable`		**Default Value:** 25%
`maxSurge`		**Default Value:** 25%
`gmlc.destIfLocQosAbsent`	Indicates the location provider when LocQos is not present in the subscription request.	This is a mandatory parameter. Possible Values are: • udm • gmlc **Default Value:** udm
`gmlc.switchToUdmOnFailure`	Enables the failover to UDM when GMLC subscription fails for a specific reason. The location reporting event request is sent to UDM when NEF receives failure from GMLC.	This is a mandatory parameter. Possible values are: • **ALL**: enables the failover for subscription of both immediate and deferred requests • **IMMEDIATE**: enables the failover for subscription of immediate location requests • **DEFERRED**: enables the failover for subscription of deferred location requests • **NONE**: disables the failover **Default Value:** true

`gmlc.switchOnErrorCodes`	Contains a list of error codes and causes for which the failover must be performed.	This is a mandatory parameter.
		Default Value: 503
		Example:
		`gmlc:`
		`switchOnErrorCodes:`
		`code:"503"`
		`cause:"Connection Refused"`
		`code:"503"`
		`cause:"Unknown Host Exception"`
		
		Note: The values for `code` and `cause` must be same as the values provided for the `global.configurableErrorCodes` parameter in the "global" section.

Table 5-9 (Cont.) Monitoring Events Parameters

Parameter	Description	Notes
`gmlc.explicitCancellation`	Indicates whether an explicit cancellation request should be sent to GMLC. • when the `maximumNumberOfReports` are received from GMLC • when Delete Subscription request is received from AF based on the GMLC initiated Cancel Location request.	This is a mandatory parameter. **Default Value:** false
`gmlc.gmlchAccuracy`	Specifies the minimum horizontal accuracy	This is a mandatory parameter. **Default Value:** 10.0
`gmlc.gmlcVaccuracy`	Specifies the minimum vertical accuracy	This is a mandatory parameter. **Default Value:** 10.0
`gmlc.reportingInterval`	The location reporting interval in seconds.	This is a mandatory parameter. **Default Value:** 3600

5.2.1.4 Quality of Service Parameters

Table 5-10 Quality of Service (QoS) Parameters

Parameter	Description	Notes
`image.name`	QoS image name	This is an optional parameter. **Default Value:** oc_nef_quality_of_service
`image.tag`	QoS image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** Always
`httpTwoEnabled`		**Default Value:** true
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
`resources.limits.hooksCpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 0.5
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi

Table 5-10 (Cont.) Quality of Service (QoS) Parameters

Parameter	Description	Notes
`resources.limits.hooksMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 0.5Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
`resources.requests.hooksCpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 0.5
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi
`resources.requests.hooksMemory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 0.5Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 80
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 1
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 1
`maxUnavailable`	Maximum number of unavailable pods for pod disruption budget and rolling update	**Default Value:** 1
`maxSurge`	Maximum extra pods spawn during rolling upgrade	
`hookRestartPolicy`	This attribute decides the restart policy of hook.	**Default Value:** Never
`podRestartPolicy`	This attribute decides the restart policy of pod.	**Default Value:** Always
`extraContainers`		Allowed Values: DISABLED, ENABLED, USE_GLOBAL_VALUE
`switchToPCRFOnAuthFailure`	This parameter needs to be set to `true`, to enable PCRF failover mechanism.	
`directPCRFFlowEnabled`	This parameter must be set to true to enable 4G only QoS subscription flow.	Allowed Values: true, false Default value: false
`switchOnErrorCodes.code`	This parameter is mapped to the error code upon which switch over shall be triggered in NEF.	
`switchOnErrorCodes.cause`	This parameter is mapped to the error cause upon which switch over shall be triggered in NEF.	

Table 5-10 (Cont.) Quality of Service (QoS) Parameters

Parameter	Description	Notes
converged.pe.charging	Enabling this parameter will trigger NEF to initiate charging request towards CHF (for all NEF to AF interactions done by QoS SVC).	Allowed values are: • true • false
loglevel.root	Log level for root logs	**Default Value:** INFO
loglevel.nef	Log level for NEF service logs	**Default Value:** INFO
loglevel.hook	Log level for hook logs	**Default Value:** INFO
jetty.serviceMeshCheck		
jetty.connectionTimeout		**Default Value:** 10000ms
jetty.requestTimeout	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 12000ms
readinessProbe.initialDelaySeconds	Informs the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
readinessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 3
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 3
livenessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
livenessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 5
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 5
service.type	Type of the service.	This is an optional parameter. **Default Value:** ClusterIP
service.customExtension.labels	Custom lables that are specific to the Qos service.	
service.customExtension.annotations	Custom annotations that are specific to the Qos service.	
deployment.customExtension.labels	Custom lables that that are specific to the deployment.	

Table 5-10 (Cont.) Quality of Service (QoS) Parameters

Parameter	Description	Notes
`deployment.customExtension.annotations`	Custom annotations that are specific to the deployment.	
`maxQosReferencesAllowed`	The maximum number of QoS references allowed.	**Default Value:** 2

5.2.1.5 Traffic Influence Parameters

Table 5-11 Traffic Influence Parameters

Parameter	Description	Notes
`image.name`	Traffic Influence image name	This is an optional parameter. **Default Value:** oc_nef_quality_of_service
`image.tag`	Traffic Influence image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** Always
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
`resources.limits.hooksCpu`	Maximum amount of CPU that Kubernetes allows the hooks to use.	This is an optional parameter. **Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi
`resources.limits.hooksmemory`	Maximum amount of memory that Kubernetes allows the hook to use.	This is an optional parameter. **Default Value:** 1
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
`resources.requests.hookscpu`	The amount of hooks CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi
`resources.requests.hooksmemory`	The amount of hooks memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi

Table 5-11 (Cont.) Traffic Influence Parameters

Parameter	Description	Notes
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 80
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 1
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 1
`readinessProbe.initialDelaySeconds`	Informs the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 40
`readinessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 3
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 3
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 40
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 5
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 5
`extraContainers`		
`deployment.customExtension.labels`	Custom lables that that are specific to the deployment.	
`deployment.customExtension.annotations`	Custom annotations that are specific to the deployment.	
`service.type`	Type of the service.	This is an optional parameter. **Default Value:** ClusterIP
`service.customExtension.labels`	Custom lables that are specific to the Qos service.	
`service.customExtension.annotations`	Custom annotations that are specific to the Qos service.	
`jetty.serviceMeshCheck`	Load balancing will be handled by Ingress gateway, if true it would be handled by serviceMesh	

Table 5-11 (Cont.) Traffic Influence Parameters

Parameter	Description	Notes
`jetty.connectionTimeout`	Jetty client connection timeout settings	
`jetty.requestTimeout`	Jetty client request timeout settings. Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	
`loglevel.root`	Log level for root logs	**Default Value:** WARN
`loglevel.nef`	Log level for NEF service logs	**Default Value:** WARN
`loglevel.hook`	Log level for hook service logs	**Default Value:** WARN

5.2.1.6 Fivegcagent Service Parameters

Table 5-12 Fivegcagent Service Parameters

Parameter	Description	Notes
`image.name`	fivegcagent image name	This is an optional parameter. **Default Value:** oc_nef_5gcagent
`image.tag`	fivegcagent image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`udmBaseUrl`	The base URL of UDM. **Note:** If mtls is enabled, use https://ocnefsim-ocstub-svc-udm:1010 url. If not, use http://ocnefsim-ocstub-svc-udm:1010.	This is a Model A parameter.
`pcfBaseUrl`	The base URL of PCF. **Note:** If mtls is enabled, use https://ocnefsim-ocstub-svc-pcf:1010 url. If not, use http://ocnefsim-ocstub-svc-pcf:1010.	This is a Model A parameter.
`bsfEnabled`	BSF to be enabled or disabled.	This is a Model A parameter.
`bsfBaseUrl`	The base URL of BSF. **Note:** If mtls is enabled, use https://ocnefsim-ocstub-svc-bsf:1010. If not, use http://ocnefsim-ocstub-svc-bsf:1010.	This is a Model A parameter.
`udrBaseUrl`	The base URL of UDR. **Note:** If mtls is enabled, use https://ocnefsim-ocstub-svc-udr:1010 url. If not, use http://ocnefsim-ocstub-svc-udr:1010.	This is a Model A parameter.
`chfBaseUrl`	The base URL of UDR.	

Table 5-12 (Cont.) Fivegcagent Service Parameters

Parameter	Description	Notes
`jetty.serviceMeshCheck`	Enable this parameter if NEF is deployed in serviceMesh environment.	
`jetty.connectionTimeout`	Jetty connection timeout to the backend peer.	
`jetty.requestTimeout`	Request timeout to the backend peer. Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2000ms
`log.level.root`	Log level for root logs	
`log.level.nef`	Log level for nef logs	
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 60
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 12
`maxUnavailable`	Max Unavailable pods for pod disruption budget and rolling update.	
`maxSurge`	Max extra pods spawn during rolling upgrade.	
`gmlc.baseUrl`	The base URL of GMLC	
`gmlc.externalClientType`	Default value to be sent to GMLC in `externalClientType` parameter in `ProvideLocation` Request.	**Default Value:** PLMN_OPERATOR_SERVICES
`gmlc.reportingInterval`	Indicates the time interval between each event report in seconds.	
Communication Profile Configuration Use this configuration to create the Model D communication profiles. The `<customModelD>` is a variable for the communication profile name. You can change the name of the profile and create multiple profiles.		

Table 5-12 (Cont.) Fivegcagent Service Parameters

Parameter	Description	Notes
`communicationProfiles.<customModelD>.discoveryHeaderParams.targetNfType`	The target NF, with which NEF is going to have the indirect communication (Model D). This parameter is mapped with the `3gpp-Sbi-Discovery-service-names discovery` header.	Possible values for <customModelD> are: • UDM • GMLC • BSF • PCF • UDR • CHR
`communicationProfiles.<customModelD>.discoveryHeaderParams.discoveryServices`	The service names for the discovery NF. This parameter is mapped with the `3gpp-Sbi-Discovery-service-names` header.	Possible values for <customModelD> are: • UDM • GMLC • BSF • PCF • UDR • CHR
`communicationProfiles.<customModelD>.discoveryHeaderParams.supportedFeatures`	This parameter is mapped with the `3gpp-Sbi-Discovery-supported-features discovery` header.	Possible values for <customModelD>are: • UDM • GMLC • BSF • PCF • UDR • CHR
`communicationProfiles.<customModelD>.discoveryHeaderParams.preferredLocality`	It is the preferred target NF location. For example, geographic location or data center.	Possible values for <customModelD> are: • UDM • GMLC • BSF • PCF • UDR • CHR
`communicationProfiles.<customModelD>.sendDiscoverHeaderInitMsg`	Flag to control whether to send discovery headers in initial message or not.	Possible values for <customModelD> are: • UDM • GMLC • BSF • PCF • UDR • CHR
`communicationProfiles.<customUDMModelD>.sendDiscoverHeaderSubsMsg`	Flag to control whether to send discovery headers in subsequent message or not.	Possible values for <customModelD> are: • UDM • GMLC • BSF • PCF • UDR • CHR

Table 5-12 (Cont.) Fivegcagent Service Parameters

Parameter	Description	Notes
`communicationProfiles.<customUDMModelD>.sendRoutingBindingHeader`	Indicates if the routing binding header must be included or not.	Possible values for <customModelD> are: • UDM • GMLC • BSF • PCF • UDR • CHR
`targetNfCommunicationProfileMapping.UDM`	The supported communication method for UDM.	Possible values are: • model A • model B • The communication profile name `<customUDMModelD>` set in `communicationProfiles` configuration.
`targetNfCommunicationProfileMapping.GMLC`	The supported communication method for GMLC.	Possible values are: • model A • model B • The communication profile name from `communicationProfiles` configuration
`targetNfCommunicationProfileMapping.PCF`	The supported communication method for PCF.	Possible values are: • model A • model B • The communication profile name from `communicationProfiles` configuration
`targetNfCommunicationProfileMapping.BSF`	The supported communication method for BSF.	Possible values are: • model A • model B • The communication profile name from `communicationProfiles` configuration
`targetNfCommunicationProfileMapping.CHF`	The supported communication method for CHF.	Possible values are: • model A • model B • customCHFModelD • The communication profile name from `communicationProfiles` configuration

Table 5-12 (Cont.) Fivegcagent Service Parameters

Parameter	Description	Notes
`fivegcagent.targetNfCommunicationProfileMapping.<customModelID>`	The supported communication method for UDR.	Possible values are: • model A • model B • model D Here, <customModelID> can be: • UDM • GMLC • BSF • PCF • UDR

5.2.1.7 ExpiryAuditor Service Parameters

Table 5-13 ExpiryAuditor Service Parameters

Parameter	Description	Notes
`image.name`	expiryAuditor image name	This is an optional parameter. **Default Value:** oc_nef_expiry_auditor
`image.tag`	expiryAuditor image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
`target.averageCpuUtil`		**Default Value:** 60
`log.level.root`	Log level for root logs	
`log.level.nef`	Log level for nef logs	

5.2.1.8 CCF Client Service Parameters

Table 5-14 CCF Client Service Parameters

Parameter	Description	Notes
nefApiroot	The API root for the CCF Client	nef-apiroot
providerRegistration.regSec	The registration ID of provider.	RegSec **Note:** This parameter value must be unique for every NEF instance.
log.level.root	Log level for root logs	
log.level.nef	Log level for nef logs	
log.level.hook	Log level for db hook logs	
image.name	ocnef-ccfclient image name	This is an optional parameter. **Default Value:** oc_nef_ccfclient_manager
image.tag	ocnef-ccfclient image tag	This is an optional parameter. **Default Value:** NEF Images
image.pullPolicy	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** Always
resources.limits.cpu	Maximum amount of CPU that Kubernetes allows the job resource to use.	0.7
resources.limits.hooksCpu	Maximum amount of CPU for hook that Kubernetes allows the job resource to use.	0.5
resources.limits.memory	Maximum amount of memory that Kubernetes allows the job resource to use.	0.7Gi
resources.limits.hooksMemory	Maximum amount of memoey for hook that Kubernetes allows the job resource to use.	0.5Gi
resources.requests.cpu	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	0.7
resources.requests.hooksCpu	The amount of hook CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	0.5
resources.requests.memory	The amount of hook memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	0.7Gi
resources.requests.hooksMemory	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	0.5Gi

Table 5-14 (Cont.) CCF Client Service Parameters

Parameter	Description	Notes
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	80
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	
`readinessProbe.initialDelaySeconds`	Informs the kubelet that it should wait xx second before performing the first probe	40
`readinessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	10
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	5
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	1
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	5
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	40
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	10
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	5
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	5
`service.type`	The service type.	
`service.customExtension.labels`	Custom lables that are specific to the CCF Client service.	
`service.customExtension.annotations`	Custom annotations that are specific to the CCF Client service.	
`deployment.customExtension.labels`	Custom lables that are specific to the deployment.	
`deployment.customExtension.annotations`	Custom annotations that are specific to the deployment.	
`extraContainers`	Specifies if extra container must be used for DEBUG tool.	Possible Values are: • USE_GLOBAL_VALUE • DISABLED • ENABLED This is an optional parameter. **Default Value:** USE_GLOBAL_VALUE

5.2.1.9 NRF Client Parameter

Table 5-15 NRF Client Parameter

Parameter	Description	Notes
`helmBasedConfigurationNodeSelectorApiVersion`	Specifies the API version for helm based configurations.	
`configmapApplicationConfig`	This config map is used to provide inputs to	
`configmapApplicationConfig.profile`	Contains configuration parameters that goes into nrf-client's config map	Refer config-map table for configurable parameters.
`configmapApplicationConfig.profile.primaryNrfApiRoot`	Primary NRF hostname and port <Hostname/IP>:<Port>	For Example: nrf1-api-gateway.svc:80
`configmapApplicationConfig.profile.SecondaryNrfApiRoot`	secondary NRF hostname and port <Hostname/IP>:<Port>	For Example: nrf2-api-gateway.svc:80
`configmapApplicationConfig.profile.nrfScheme`	**Note:** If mtls is enabled, use https:// in url.	In Model B, NRFclient parameters are used to identify mtls.
`configmapApplicationConfig.profile.retryAfterTime`	When primary NRF is down, this will be the wait Time (in ISO 8601 duration format) after which request to primary NRF will be retried to detect primary NRF's availability.	For Example: PT120S
`configmapApplicationConfig.profile.nrfClientType`	The NfType of the NF registering. This should be set to NEF.	
`configmapApplicationConfig.profile.nrfClientSubscribeTypes`	NF Type(s) for which the NF wants to discover and subscribe to the NRF.	Leave blank if NEF does not require. Allowed values: • UDM • UDR • PCF • BSF • CHF
`configmapApplicationConfig.profile.appProfiles`	NfProfile of NEF to be registered with NRF.	It is a 3GPP defined data type. To know more about its attributes, refer to *3GPP TS 29.510 version 16.4.0 Release 16*.
`configmapApplicationConfig.profile.enableF3`	Support for 29.510 Release 15.3	
`configmapApplicationConfig.profile.enableF5`	Support for 29.510 Release 15.5	
`configmapApplicationConfig.profile.registrationRetryInterval`		
`configmapApplicationConfig.profile.subscriptionRetryInterval`		

Table 5-15 (Cont.) NRF Client Parameter

Parameter	Description	Notes
`configmapApplicationConfig.profile.discoveryRetryInterval`		
`configmapApplicationConfig.profile.renewalTimeBeforeExpiry`	Time Period(seconds) before the Subscription Validity time expires	For Example: 3600 (1hr)
`configmapApplicationConfig.profile.validityTime`	The default validity time(days) for subscriptions	For Example: 30 (30 days)
`configmapApplicationConfig.profile.enableSubscriptionAutoRenewal`	Enable Renewal of Subscriptions automatically	
`configmapApplicationConfig.profile.nfHeartbeatRate`	The default rate at which the NF shall heartbeat with the NRF. The value shall be configured in terms of percentage(1-100). If the heartbeatTimer is 60s, then the NF shall heartbeat at nfHeartBeatRate * 60/100	
`configmapApplicationConfig.profile.acceptAdditionalAttributes`	Enable additional Attributes as part of 29.510 Release 15.5	
`configmapApplicationConfig.profile.retryForCongestion`		
`configmapApplicationConfig.profile.supportedDataSetId`		
`configmapApplicationConfig.profile.enableVirtualNrfResolution`	enable virtual NRF session retry by Alternate routing service	
`configmapApplicationConfig.profile.virtualNrfFqdn`	virtual NRF FQDN used to query static list of route	
`configmapApplicationConfig.profile.virtualNrfScheme`	Scheme to be used with the virtual FQDN	
`configmapApplicationConfig.profile.useAlternateScpOnAlternateRouting`		
`configmapApplicationConfig.profile.subscriberNotificationRetry`		
`configmapApplicationConfig.profile.requestTimeoutGracePeriod`	An additional grace period where no response is received from the NRF.This additional period shall be added to the requestTimeout value.This will ensure that the egress-gateway shall first timeout, and send an error response to the	

Table 5-15 (Cont.) NRF Client Parameter

Parameter	Description	Notes
`configmapApplicationConfig.profile.nrfRetryConfig`	Specifies all the parameters that need to be configured for NRF retry in case of failures per Service Request Type.	
`configmapApplicationConfig.profile.healthCheckConfig`	Specifies the parameters that need to be configured for the HealthCheck mechanism.	
nrf-client-nfmanagement		
`nrf-client-nfmanagement.image`		nrf-client
`nrf-client-nfmanagement.imageTag`		
`nrf-client-nfmanagement.dbEngine`		
`nrf-client-nfmanagement.dbConfig.dbHost`		
`nrf-client-nfmanagement.dbConfig.dbPort`		
`nrf-client-nfmanagement.dbConfig.dbName`		
`nrf-client-nfmanagement.dbConfig.leaderPodDbName`		
`nrf-client-nfmanagement.dbConfig.networkDbName`		
`nrf-client-nfmanagement.dbConfig.secretName`		
`nrf-client-nfmanagement.dbConfig.dbUNameLiteral`		
`nrf-client-nfmanagement.dbConfig.dbPwdLiteral`		
nrf-client-nfdiscovery		
`nrf-client-nfdiscovery.image`		nrf-client
`nrf-client-nfdiscovery.imageTag`		
`nrf-client-nfdiscovery.dbConfig.dbEngine`		
`nrf-client-nfdiscovery.dbConfig.dbHost`		

Table 5-15 (Cont.) NRF Client Parameter

Parameter	Description	Notes
nrf-client-nfdiscovery.dbConfig.dbPort		
nrf-client-nfdiscovery.dbConfig.dbName		
nrf-client-nfdiscovery.dbConfig.secretName		
nrf-client-nfdiscovery.dbConfig.dbUNameLiteral		
nrf-client-nfdiscovery.dbConfig.dbPwdLiteral		
config-server		
config-server.nfInstanceId	NF InstanceId of service consumer	
config-server.image	config-server image name	This is an optional parameter. **Default Value:** oc-config-server
config-server.imageTag	Tag name of image	This is an optional parameter. **Default Value:** NEF Images
config-server.envMysqlDatabase	Name of the database for Config Server service	
config-server.dbConfig.dbEngine	The database engine name	
config-server.resources.limits.ephemeralStorage	Indicates the minimum limit of ephemeral-storage	
config-server.resources.limits.cpu	Maximum amount of CPU that Kubernetes allows the job resource to use.	
config-server.resources.limits.memory	Maximum amount of memory that Kubernetes allows the job resource to use.	
config-server.requests.requests.cpu	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	
config-server.requests.requests.memory	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	
appinfo		
appinfo.image	config-server image name	This is an optional parameter. **Default Value:** oc-app-info

Table 5-15 (Cont.) NRF Client Parameter

Parameter	Description	Notes
appinfo.imageTag	Tag name of image	This is an optional parameter. **Default Value:** NEF Images
appinfo.core_services.nef	Specifies the list of NEF services to be monitored.	**Syntax:** *<Helm Release Name>-<Service Name>* **Sample Value:** ocnef-monitoringevents
appinfo.dbConfig.dbHost	Hostname of database connection in which the common configuration must be stored	
appinfo.dbConfig.dbPort	Port of database connection	
appinfo.dbConfig.dbName	Name of common configuration database	
appinfo.dbConfig.secretName	Secret name from which the DB name, DB password and db user name is picked for common configuration	
appinfo.dbConfig.dbUNameLiteral	Name of the Key configured for "DB Username" in Secret with following name: "<dbConfig.secretName>"	
appinfo.dbConfig.dbPwdLiteral	Name of the Key configured for "DB Password" in Secret with following name: "<dbConfig.secretName>"	
perf-info		
perf-info.image	perf-server image name	This is an optional parameter. **Default Value:** oc-perf-info
perf-info.imageTag	Tag name of image	This is an optional parameter. **Default Value:** NEF Images
perf-info.nfType	Specifies the list of NEF services to be monitored.	
perf-info.configmapPerformance.prometheus	Specifies Prometheus server URL	
perf-info.dbConfig.dbHost	Hostname of database connection in which the common configuration must be stored	
perf-info.dbConfig.dbPort	Port of database connection	
perf-info.dbConfig.dbName	Name of common configuration database	
perf-info.dbConfig.secretName	Secret name from which the DB name, DB password and db user name is picked for common configuration	
perf-info.dbConfig.dbUNameLiteral	Name of the Key configured for "DB Username" in Secret with following name: "<dbConfig.secretName>"	
perf-info.dbConfig.dbPwdLiteral	Name of the Key configured for "DB Password" in Secret with following name: "<dbConfig.secretName>"	

5.2.1.10 AEF API Router Parameters

Table 5-16 AEF API Router Parameters

Parameter	Description	Notes
`enabled`	Specifies if the service is enabled.	**Default Value:** true
`image.name`	Name of image.	This is an optional parameter. **Default Value:** oc_nef_aef_apirouter
`image.tag`	Tag name of image.	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`certificate.secretName`	Secret name that contains NEF's certificate for HTTPS	certificate-secret
`certificate.secretNamespace`	Namespace in which k8SecretName is present	
`certificate.certificateName`	NEF's Certificate (RSA type) file name	tmp.cer **Note:** The certificate name must be same as the external gateway certificate name for CAPIF. For more information, see the `externalGWConfig.tls.certificate.rsa.filename` parameter in Global Parameters.
`certificate.publicKeyMonitorDelay`	NEF's Certificate (ECDSA type) file name	15000
`invokerInfoCache.expiry`		
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
`resources.limits.initServiceCpu`		This is an optional parameter. **Default Value:** 1
`resources.limits.updateServiceCpu`		This is an optional parameter. **Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi
`resources.limits.initServiceMemory`		This is an optional parameter. **Default Value:** 1Gi
`resources.limits.updateServiceMemory`		This is an optional parameter. **Default Value:** 1Gi

Table 5-16 (Cont.) AEF API Router Parameters

Parameter	Description	Notes
resources.requests.cpu	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
resources.requests.initServiceCpu		This is an optional parameter. **Default Value:** 1
resources.requests.updateServiceCpu		This is an optional parameter. **Default Value:** 1
resources.requests.memory	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi
resources.requests.initServiceMemory		This is an optional parameter. **Default Value:** 1Gi
resources.requests.updateServiceMemory		This is an optional parameter. **Default Value:** 1Gi
resources.target.averageCpuUtil	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	This is an optional parameter. **Default Value:** 80
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 1
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 3
maxUnavailable		This is an optional parameter. **Default Value:** 1
hookRestartPolicy		Never
podRestartPolicy		Always
readinessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
readinessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 3
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 3
livenessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
livenessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10

Table 5-16 (Cont.) AEF API Router Parameters

Parameter	Description	Notes
livenessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 5
livenessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
livenessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 5
deployment.customExtension.labels	Custom lables that needs to be added to API Router deployment.	
deployment.customExtension.annotations	Custom annotations that needs to be added to API Router deployment.	
initssl	Specifies if the key and truststore have to be generated.	This value must always be set as true. **Default Value:** true
service.type	The type of the service.	
service.customExtension.labels	Custom labels that needs to be added to API Router service.	
service.customExtension.annotations	Custom annotations that needs to be added to API Router service.	
extraContainers	Specifies if extra container must be used for DEBUG tool.	
log.level.root	Log level for root logs	
log.level.nef	Log level for nef service logs	
log.level.updateContainer	Log level for updateContainer logs	
application.threadpool.size		**Default Value:** 2300 ms
application.threadpool.queue		
application.threadpool.timeout		
backend.serviceMeshCheck		
concurrentPushedStreams		
queuedPerDes		
connectionsPerDest		
connectionsPerIp		
requestTimeOut	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2500 ms
threadpool.size		
threadpool.queue		
server.threadpool.size		

Table 5-16 (Cont.) AEF API Router Parameters

Parameter	Description	Notes
coherence.messagingPort1		
coherence.messagingPort2		
terminationGracePeriodSeconds	This parameter provides time to transfer the Coherence cache data in cluster. This value can be changed as per the server slowness.	**Default Value:** 45 seconds

5.2.1.11 Ingress Gateway Parameters

The following table describes the parameters for the 5GC Ingress GW and External Ingress GW services.

- FiveGC Ingress GW
- External Ingress GW

FiveGC Ingress GW

Table 5-17 NEF FiveGC Ingress Gateway (ingress-gateway) Parameters

Parameter	Description	Details
global.publicHttpSignalingPort	HTTP service port on which NEF Ingress Gateway is exposed	
global.publicHttpsSignallingPort	HTTPS service port on which NEF Ingress Gateway is exposed	
serviceAccountName		
global.type	The service type that will be used for this deployment.	It is not recommended to change the service type.
global.staticIpAddressEnabled	Specifies if static load balancer IP needs to be set	**Default Value:** false
global.staticIpAddress	Static IP address assigned to the Load Balancer from the external load balancer IP pool.	
global.staticNodePortEnabled	Specifies if static node port needs to be set	**Default Value:** false
global.staticHttpNodePort	Static HTTP Node Port	**Default Value:** 30075
global.staticHttpsNodePort	Static HTTPS Node Port	**Default Value:** 30043

Table 5-17 (Cont.) NEF FiveGC Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`image.name`	Ingress Gateway image name	This is an optional parameter. **Default Value:** ocingress_gateway
`image.tag`	Ingress Gateway image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`initContainersImage.name`	Image Name for Ingress GW init container	This is an optional parameter. **Default Value:** configurationinit
`initContainersImage.tag`	Tag Name for Ingress Gateway init container	This is an optional parameter. **Default Value:** NEF Images
`initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`updateContainersImage.name`	Image Name for Ingress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
`updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** NEF Images
`updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`extraContainers`	Specifies if extra container must be used for DEBUG tool.	
`service.ssl.keyType`	The selected key type.	
`service.ssl.tlsVersion`	The TLS version.	
`service.ssl.privateKey.k8SecretName`	Secret name that contains NEF Ingress Gateway Private Key	
`service.ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	

Table 5-17 (Cont.) NEF FiveGC Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`service.ssl.privateKey.rsa.filename`	NEF's Private Key (RSA type) file name	
`service.ssl.privateKey.ecdsa.filename`	NEF Ingress Gateway Private Key (ecdsa type) file name	
`service.certificate.k8SecretName`	Secret name that contains NEF Ingress Gateway certificate for HTTPS	
`service.certificate.k8NameSpace`	Namespace in which k8SecretName is present	
`service.certificate.rsa.filename`	NEF Ingress Gateway Certificate (RSA type) file name	
`service.certificate.ecdsa.filename`	NEF Ingress Gateway Certificate (ECDSA type) file name	
`service.caBundle.k8SecretName`	Secret name that contains NEF Ingress Gateway's CA details for HTTPS	
`service.caBundle.k8NameSpace`	Namespace that contains NEF Ingress Gateway's CA details for HTTPS	
`caBundle.filename`	NEF Ingress Gateway's CA bundle filename	
`service.keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	
`service.keyStorePassword.k8NameSpace`	Namespace in which NEF Ingress Gateway's keystore password is present	
`service.keyStorePassword.filename`	NEF Ingress Gateway's Key Store password Filename	
`service.trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	NEF Ingress Gateway's trustStorePassword Filename	O
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway deployment.	

Table 5-17 (Cont.) NEF FiveGC Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`log.level.root`	Log level for root logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.ingress`	Log level for ingress logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.oauth`	Log level for oauth logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.updateContainer`	Log level for updateContainer logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.hook`	Log level for hook logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE

Table 5-17 (Cont.) NEF FiveGC Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`log.level.cncc.security`	Log level for CNC Console security	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.traceIdGenerationEnabled`		**Default value:** true
`requestTimeOut`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 2500 ms
`startupProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the startup probe	**Default Value:** 30
`startupProbe.periodSeconds`	specifies that the kubelet should perform a startupProbe probe every xx seconds	**Default Value:** 3
`startupProbe.timeoutSeconds`		**Default Value:** 10
`startupProbe.successThreshold`		**Default Value:** 1
`startupProbe.failureThreshold`		**Default Value:** 6
`readinessProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`readinessProbe.periodSeconds`	Specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 10
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 15

Table 5-17 (Cont.) NEF FiveGC Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.commonHooksCpu`	Maximum amount of commonHooksCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.commonHooksMemory`	Maximum amount of commonHooksMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.commonHooksCpu`	Maximum amount of commonHooksCpu that Kubernetes allows the job resource to use.	**Default Value:** 1

Table 5-17 (Cont.) NEF FiveGC Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.commonHooksMemory`	The amount of commonHooksMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 60
`minAvailable`	Number of Pods that must always be available, even during a disruption.	**Default Value:** 2
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5
`metricPrefix`	If this value is configured then the value is added as a prefix to all the metrics in the ingress gateway.	
`metricSuffix`	If this value is configured then the value is added as a suffix to all the metrics in the ingress gateway.	
`jaegerTelemetryTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing at Ingress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
`openTelemetry.jaeger.httpExporter.host`	Specifies the host of Jaeger collector service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger collector service	**Default Value:** 4318
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5

Table 5-17 (Cont.) NEF FiveGC Ingress Gateway (`ingress-gateway`) Parameters

Parameter	Description	Details
initssl	Specifies if SSL related infrastructure in init/update containeris to be Initialized.	This value must always be set as true. **Default Value:** true
enableIncomingHttp	This flag is for enabling/disabling HTTP/2.0 (insecure) in Ingress Gateway.	If the value is set to false, EG will not accept any HTTP/2.0 (insecure) Traffic
enableIncomingHttps	This flag is for enabling/disabling HTTPS/2.0 (secure) in Ingress Gateway.	If the value is set to false, EG will not accept any HTTPS/2.0 (secure) Traffic
needClientAuth	Specifies if client certificate identity is required in the header x-custom-ingress-client-identity	
dnsRefreshDelay	Dns Refresh Delay in milli-seconds	**Default Value:** 5000
pingDelay	Delay between pings in seconds.When set to <=0,ping is disabled	
globalRemoveRequestHeader	Attribute for blocklisting (removing) a request header at global level.	
globalRemoveResponseHeader	Attribute for blocklisting (removing) a response header at global level.	
requestContentMetricsEnabled	Attribute for enabling/disabling requestContentMetrics at global level. If enabled, will be applied to all routes configured.	
tolerations		
helmBasedConfigurationNodeSelectorApiVersion	REST API version for helm based configurations	**Default Value:** v1
nodeSelector.nodeKey		
nodeSelector.nodeValue		
ingressServer.keepAlive.enabled	Specifies if keep alive is to be enabled.	
ingressServer.keepAlive.idealTime	The keep alive ideal time settings	
ingressServer.keepAlive.count	The keep alive count settings	
ingressServer.keepAlive.interval	The keep alive interval settings	

External Ingress GW

Table 5-18 NEF External Ingress Gateway (`nef-ingress-gateway`) Parameters

Parameter	Description	Details
`global.type`	The service type that will be used for this deployment.	It is not recommended to change the service type. **Default Value:** LoadBalancer
`global.staticIpAddressEnabled`	Specifies if static load balancer IP needs to be set	**Default Value:** false
`global.staticIpAddress`	Static IP address assigned to the Load Balancer from the external load balancer IP pool.	
`global.staticNodePortEnabled`	Specifies if static node port needs to be set	**Default Value:** false
`global.staticHttpNodePort`	Static HTTP Node Port	
`global.staticHttpsNodePort`	Static HTTPS Node Port	
`global.publicHttpSignalingPort`	HTTP service port on which External Ingress Gateway is exposed	
`global.publicHttpsSignallingPort`	HTTPS service port on which External Ingress Gateway is exposed	
`global.enableIncomingHttp`	This flag is for enabling or disabling HTTP/2.0 (insecure) in Ingress Gateway.	If the value is set to false, NEF will not accept any HTTP/2.0 (insecure) traffic.
`global.enableIncomingHttps`	This flag is for enabling or disabling HTTPS/2.0 (secure) in Ingress Gateway.	If the value is set to false, NEF will not accept any HTTPS/2.0 (secured) traffic.
`image.name`	Ingress Gateway image name	This is an optional parameter. **Default Value:** ocingress_gateway
`image.tag`	Ingress Gateway image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`initContainersImage.name`	Image Name for Ingress GW init container	This is an optional parameter. **Default Value:** configurationinit
`initContainersImage.tag`	Tag Name for Ingress Gateway init container	This is an optional parameter. **Default Value:** NEF Images

Table 5-18 (Cont.) NEF External Ingress Gateway (`nef-ingress-gateway`) Parameters

Parameter	Description	Details
`initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`updateContainersImage.name`	Image Name for Ingress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
`updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** NEF Images
`updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`jaegerTelemetryTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing at Ingress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
`openTelemetry.jaeger.httpExporter.host`	Specifies the host of Jaeger collector service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger collector service	**Default Value:** 4318
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5
`service.ssl.tlsVersion`	The TLS version.	
`service.ssl.privateKey.k8SecretName`	Secret name that contains NEF Ingress Gateway Private Key	
`service.ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	
`service.ssl.privateKey.rsa.filename`	NEF's Private Key (RSA type) file name	
`service.ssl.privateKey.ecdsa.filename`	NEF Ingress Gateway Private Key (ecdsa type) file name	
`service.certificate.k8SecretName`	Secret name that contains NEF Ingress Gateway certificate for HTTPS	
`service.certificate.k8NameSpace`	Namespace in which k8SecretName is present	
`service.certificate.rsa.filename`	NEF Ingress Gateway Certificate (RSA type) file name	

Table 5-18 (Cont.) NEF External Ingress Gateway (`nef-ingress-gateway`) Parameters

Parameter	Description	Details
`service.certificate.ecdsa.filename`	NEF Ingress Gateway Certificate (ECDSA type) file name	
`service.caBundle.k8SecretName`	Secret name that contains NEF Ingress Gateway's CA details for HTTPS	
`service.caBundle.k8NameSpace`	Namespace that contains NEF Ingress Gateway's CA details for HTTPS	
`caBundle.filename`	NEF Ingress Gateway's CA bundle filename	
`service.keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	
`service.keyStorePassword.k8NameSpace`	Namespace in which NEF Ingress Gateway's keystore password is present	
`service.keyStorePassword.filename`	NEF Ingress Gateway's Key Store password Filename	
`service.trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	NEF Ingress Gateway's trustStorePassword Filename	O
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Ingress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Ingress Gateway deployment.	
`log.level.root`	Log level for root logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE

Table 5-18 (Cont.) NEF External Ingress Gateway (`nef-ingress-gateway`) Parameters

Parameter	Description	Details
`log.level.ingress`	Log level for ingress logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.oauth`	Log level for oauth logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.updateContainer`	Log level for updateContainer logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.configclient`	Log level for configclient logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`log.level.hook`	Log level for hook logs	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE

Table 5-18 (Cont.) NEF External Ingress Gateway (`nef-ingress-gateway`) Parameters

Parameter	Description	Details
`log.level.cncc.security`	Log level for CNC Console security	Possible values are: • OFF • FATAL • ERROR • WARN • INFO • DEBUG • TRACE
`startupProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`startupProbe.periodSeconds`	Specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`startupProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 10
`startupProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`startupProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`readinessProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`readinessProbe.periodSeconds`	Specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 10
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 15
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1

Table 5-18 (Cont.) NEF External Ingress Gateway (`nef-ingress-gateway`) Parameters

Parameter	Description	Details
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi

Table 5-18 (Cont.) NEF External Ingress Gateway (`nef-ingress-gateway`) Parameters

Parameter	Description	Details
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 80
`requestTimeOut`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 15000 ms
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5

5.2.1.12 Egress Gateway Parameters

The following table describes the parameters for the FiveGC Egress Gateway and External Egress Gateway services.

- FiveGC Egress GW
- External Egress GW

FiveGC Egress GW

Table 5-19 FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`serviceEgressGateway.port`	The Egress Gateway service port	
`deploymentEgressGateway.image`	egress-gateway image name	This is an optional parameter. **Default Value:** ocegress_gateway
`deploymentEgressGateway.imageTag`	egress-gateway image tag	This is an optional parameter. **Default Value:** NEF Images
`deploymentEgressGateway.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`deploymentEgressGateway.initContainersImage.name`	Image Name for Egress Gateway init container	This is an optional parameter. **Default Value:** configurationinit
`deploymentEgressGateway.initContainersImage.tag`	Tag Name for Egress Gateway init container	This is an optional parameter. **Default Value:** NEF Images
`deploymentEgressGateway.initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`deploymentEgressGateway.updateContainersImage.name`	Image Name for Egress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
`deploymentEgressGateway.updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** NEF Images
`deploymentEgressGateway.updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`extraContainers`	Specifies if extra container must be used for DEBUG tool.	
`initssl`		
`enableOutgoingHttps`	This flag is for enabling/disabling HTTPS/2.0 (secured TLS) in Egress Gateway.	Set this value to true only if the initssl is set to true.
`pingDelay`	Delay between pings in seconds.When set to <=0,ping is disabled	
`startupProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the startup probe	
`startupProbe.periodSeconds`	specifies that the kubelet should perform a startupProbe probe every xx seconds	
`startupProbe.timeoutSeconds`		
`startupProbe.successThreshold`		
`startupProbe.failureThreshold`		

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`readinessProbe.initialDelaySeconds`		
`readinessProbe.periodSeconds`	specifies that the kubelet should perform a readinessProbe probe every xx seconds	
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	
SbiRouting Configuration Use this configurations for SCP integration with NEF including SBI routing.		
`sbiRoutingDefaultScheme`	Specifies the default scheme applicable when `3gpp-sbi-target-apiroot` header is missing.	**Default value:** http **Note:** If required this parameter can be configured to https.
`peerConfiguration`	Configurations for the list of peers. Each peer must contain the following: • id • host • port • apiPrefix	You can create multiple peers using `peerConfiguration`.

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`peerSetConfiguration`	Configurations for the list of peer sets. Each peer set must contain the following: • id • httpConfiguration • httpsConfiguration	Each peer set must contain HTTP or HTTPS instances where in each instance contains priority and peer identifier, which maps to peers configured under `peerConfiguration` No two instances should have same priority for a given HTTP or HTTPS configuration. In addition, more than one virtual FQDN should not be configured for a given HTTP or HTTPS configuration.
`routesConfig.id`	Specifies the ID of the route.	
`routesConfig.uri`	Provide any dummy url, or leave the existing url with existing value	
`routesConfig.path`	Specifies the path to be matched	
`routesConfig.order`	Specifies the order of the execution of this route.	
`routesConfig.metadata.httpRuriOnly`	Flag to enable httpRuriOnly functionality. When value is set to true, the RURI scheme is changed to http. For the value given as false, no changes are made to the scheme.	This value must be **true** for enabling SBIRouting.
`routesConfig.metadata.httpsTargetOnly`	Flag to enable httpsTargetOnly functionality. When the value is set to true, SBI instances are selected for HTTPS list only (if 3gpp sbitarget root header is http). When the value is set to false, no changes are made to the existing scheme.	This value must be **true** for enabling SBIRouting.
`routesConfig.metadata.sbiRoutingEnabled`	Flag to enable the sbiRouting for the selected route.	This value must be **true** for enabling SBIRouting.
FilterName configurations Use the `filterName1` configurations to configure SBIRouting filters. **Note:** To enable SBI routing, the values of all the flags in `routesConfig.metadata` must be set to **true** and sbiRouting `filterName1` must be configured.		
`routesConfig.filterName1.name`	Name of the SBIRouting filter.	Provide name as `SBIRouting`.
`routesConfig.filterName1.args.peerSetIdentifier`	Specifies the ID of the peerSetConfiguration.	

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`routesConfig.filterName 1.args.customPeerSelect orEnabled`	This flag allows the user to send request to a particular instance directly when enabled according to "oc-alternateroute-attempt" header".	**Default value:** false
`routesConfig.filterName 1.args.errorHandling`	The errorHandling section contains an array of errorcriteriaset and actionset mapping with priority. The errorcriteriaset and actionset are configured through Helm using `sbiRoutingErrorCriteriaSets` and `sbiRoutingErrorActionSets`.	To disable the rerouting under SBIRouting, delete the `errorHandling` configurations under `routesConfig`.
`routesConfig.filterName 1.args.errorHandling.ac tionSet`	Contains an array of actionset, where each depicts an ID, action to be performed (Currently on REROUTE action is supported) and blocklist configurations.	
`routesConfig.filterName 1.args.errorHandling.pr iority`		

`sbiRoutingErrorCriteria Sets`	Contains an array of errorCriteriaSet , where each errorCriteriaSet depicts an ID, set of HTTP Methods, set of HTTP Response status codes, set of exceptions with headerMatching functionality.	**Example:**
		`sbiRoutingErrorCriteriaSe ts:`
		`- id: scp_direct2_criteria_1`
		`method:`
		`- GET`
		`- POST`
		`- PUT`
		`- DELETE`
		`- PATCH`
		`response:`
		`statuses:`
		`- statusSeries: 4xx`
		`status:`
		`- 400`
		`- 404`
		`- statusSeries: 5xx`
		`status:`
		`- 500`
		`- 503`

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details

`sbiRoutingErrorActionSets`	Contains an array of actionset, where each depicts an ID, action to be performed (Currently on REROUTE action is supported) and blocklist configurations.	**Example:**
		`sbiRoutingErrorActionSets:`
		`  - id: scp_direct2_action_1`
		`  action: reroute`
		`  attempts: 1`
		`  blackList:`
		`  enabled: false`
		`  duration: 60000`

`loglevel.root`	Log level for root logs	
`loglevel.egress`	Log level for egress logs	
`loglevel.oauth`	Log level for oauth logs	
`loglevel.updateContainer`	Log level for updateContainer logs	
`loglevel.hook`	Log level for hook logs	
`service.ssl.keyType`	The selected key type.	
`service.ssl.tlsVersion`	The TLS version.	
`service.ssl.privateKey.k8SecretName`	Secret name that contains NEF Ingress Gateway Private Key	
`service.ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	
`service.ssl.privateKey.rsa.filename`	NEF's Private Key (RSA type) file name	
`service.ssl.privateKey.ecdsa.filename`	NEF Egress Gateway Private Key (ecdsa type) file name	
`service.certificate.k8SecretName`	Secret name that contains NEF Egress Gateway certificate for HTTPS	
`service.certificate.k8NameSpace`	Namespace in which k8SecretName is present	
`service.certificate.rsa.filename`	NEF Egress Gateway Certificate (RSA type) file name	
`service.certificate.ecdsa.filename`	NEF Egress Gateway Certificate (ECDSA type) file name	
`service.caBundle.k8SecretName`	Secret name that contains NEF Egress Gateway's CA details for HTTPS	
`service.caBundle.k8NameSpace`	Namespace that contains NEF Egress Gateway's CA details for HTTPS	
`caBundle.filename`	NEF Egress Gateway's CA bundle filename	
`service.keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`service.keyStorePassword.k8NameSpace`	Namespace in which NEF Egress Gateway's keystore password is present	
`service.keyStorePassword.filename`	NEF Egress Gateway's Key Store password Filename	
`service.trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	NEF Egress Gateway's trustStorePassword Filename	O
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Egress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Egress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway deployment.	
`globalRemoveRequestHeader`	Attribute for blocklisting (removing) a request header at global level.	
`globalRemoveResponseHeader`	Attribute for blocklisting (removing) a response header at global level.	
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.commonHooksCpu`	Maximum amount of common hook CPU that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.commonHooksMemory`	Maximum amount of hook service memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.commonHooksCpu`	The amount of hook service CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.commonHooksMemory`	The amount of hook service memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 60
`minAvailable`	Minimum available pods	**Default Value:** 2
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2

Table 5-19 (Cont.) FiveGC Egress Gateway (`egress-gateway`) Parameters

Parameter	Description	Details
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5
`headerIndexing.doNotIndex`	The list of headers which do not require indexing.	
`requestTimeOut`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 1500 ms
`metricPrefix`	If this value is configured then the value is added as a prefix to all the metrics in the egress gateway.	
`metricSuffix`	If this value is configured then the value is added as a suffix to all the metrics in the egress gateway.	
`jaegerTelemetryTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing at Egress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
`openTelemetry.jaeger.httpExporter.host`	Specifies the host of Jaeger collector service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger collector service	**Default Value:** 4318
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5
`tolerations`		
`helmBasedConfigurationNodeSelectorApiVersion`	REST API version for helm based configurations	**Default Value:** v1
`nodeSelector.nodeKey`		
`nodeSelector.nodeValue`		

External Egress GW

Table 5-20 External Egress Gateway (`nef-egress-gateway`) Parameters

Parameter	Description	Details
`serviceEgressGateway.port`	The Egress Gateway service port	
`deploymentEgressGateway.image`	egress-gateway image name	This is an optional parameter. **Default Value:** ocegress_gateway
`deploymentEgressGateway.imageTag`	egress-gateway image tag	This is an optional parameter. **Default Value:** NEF Images

Table 5-20 (Cont.) External Egress Gateway (`nef-egress-gateway`) Parameters

Parameter	Description	Details
`deploymentEgressGateway.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`deploymentEgressGateway.initContainersImage.name`	Image Name for Egress Gateway init container	This is an optional parameter. **Default Value:** configurationinit
`deploymentEgressGateway.initContainersImage.tag`	Tag Name for Egress Gateway init container	This is an optional parameter. **Default Value:** NEF Images
`deploymentEgressGateway.initContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`deploymentEgressGateway.updateContainersImage.name`	Image Name for Egress Gateway update container	This is an optional parameter. **Default Value:** configurationupdate
`deploymentEgressGateway.updateContainersImage.tag`	Tag Name for update container	This is an optional parameter. **Default Value:** NEF Images
`deploymentEgressGateway.updateContainersImage.pullPolicy`	Image pull policy	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`extraContainers`	Specifies if extra container must be used for DEBUG tool.	
`initssl`		This value must always be set as true. **Default Value:** true
`enableOutgoingHttps`	This flag is for enabling/disabling HTTPS/2.0 (secured TLS) in Egress Gateway.	
`pingDelay`	Delay between pings in seconds.When set to <=0,ping is disabled	
`startupProbe.initialDelaySeconds`	Tells the kubelet that it should wait xx second before performing the startup probe	**Default Value:** 30

Table 5-20 (Cont.) External Egress Gateway (`nef-egress-gateway`) Parameters

Parameter	Description	Details
`startupProbe.periodSeconds`	specifies that the kubelet should perform a startupProbe probe every xx seconds	**Default Value:** 3
`startupProbe.timeoutSeconds`		**Default Value:** 10
`startupProbe.successThreshold`		**Default Value:** 1
`startupProbe.failureThreshold`		**Default Value:** 3
`readinessProbe.initialDelaySeconds`		**Default Value:** 30
`readinessProbe.periodSeconds`	specifies that the kubelet should perform a readinessProbe probe every xx seconds	**Default Value:** 3
`readinessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 10
`readinessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`readinessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`livenessProbe.initialDelaySeconds`	tells the kubelet that it should wait xx second before performing the first probe	**Default Value:** 30
`livenessProbe.periodSeconds`	specifies that the kubelet should perform a liveness probe every xx seconds	**Default Value:** 3
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	**Default Value:** 15
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	**Default Value:** 1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	**Default Value:** 3
`loglevel.root`	Log level for root logs	
`loglevel.egress`	Log level for egress logs	
`loglevel.oauth`	Log level for oauth logs	
`loglevel.updateContainer`	Log level for updateContainer logs	
`loglevel.hook`	Log level for hook logs	
`service.ssl.keyType`	The selected key type.	
`service.ssl.tlsVersion`	The TLS version.	
`service.ssl.privateKey.k8SecretName`	Secret name that contains NEF Ingress Gateway Private Key	

Table 5-20 (Cont.) External Egress Gateway (`nef-egress-gateway`) Parameters

Parameter	Description	Details
`service.ssl.privateKey.k8NameSpace`	Namespace in which k8SecretName is present	
`service.ssl.privateKey.rsa.filename`	NEF's Private Key (RSA type) file name	
`service.ssl.privateKey.ecdsa.filename`	NEF Egress Gateway Private Key (ecdsa type) file name	
`service.certificate.k8SecretName`	Secret name that contains NEF Egress Gateway certificate for HTTPS	
`service.certificate.k8NameSpace`	Namespace in which k8SecretName is present	
`service.certificate.rsa.filename`	NEF Egress Gateway Certificate (RSA type) file name	
`service.certificate.ecdsa.filename`	NEF Egress Gateway Certificate (ECDSA type) file name	
`service.caBundle.k8SecretName`	Secret name that contains NEF Egress Gateway's CA details for HTTPS	
`service.caBundle.k8NameSpace`	Namespace that contains NEF Egress Gateway's CA details for HTTPS	
`caBundle.filename`	NEF Egress Gateway's CA bundle filename	
`service.keyStorePassword.k8SecretName`	Secret name that contains keyStorePassword	
`service.keyStorePassword.k8NameSpace`	Namespace in which NEF Egress Gateway's keystore password is present	
`service.keyStorePassword.filename`	NEF Egress Gateway's Key Store password Filename	
`service.trustStorePassword.k8SecretName`	Secret name that contains trustStorePassword	
`service.trustStorePassword.k8NameSpace`	Namespace in which trustStorePassword is present	
`service.trustStorePassword.filename`	NEF Egress Gateway's trustStorePassword Filename	O
`service.initialAlgorithm`	Initial Algorithm for HTTPS	
`service.customExtension.labels`	Custom lables that needs to be added to Egress Gateway service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to Egress Gateway deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to Egress Gateway deployment.	

Table 5-20 (Cont.) External Egress Gateway (`nef-egress-gateway`) Parameters

Parameter	Description	Details
`globalRemoveRequestHeader`	Attribute for blocklisting (removing) a request header at global level.	
`globalRemoveResponseHeader`	Attribute for blocklisting (removing) a response header at global level.	
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	**Default Value:** 4
`resources.limits.initServiceCpu`	Maximum amount of initServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.updateServiceCpu`	Maximum amount of updateServiceCpu that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.commonHooksCpu`	Maximum amount of common hook CPU that Kubernetes allows the job resource to use.	**Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 4Gi
`resources.limits.initServiceMemory`	Maximum amount of memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.updateServiceMemory`	Maximum amount of updateServiceMemory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.limits.commonHooksMemory`	Maximum amount of hook service memory that Kubernetes allows the job resource to use.	**Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4
`resources.requests.initServiceCpu`	The amount of initServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.updateServiceCpu`	The amount of updateServiceCpu that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1
`resources.requests.commonHooksCpu`	The amount of hook service CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1

Table 5-20 (Cont.) External Egress Gateway (`nef-egress-gateway`) Parameters

Parameter	Description	Details
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 4Gi
`resources.requests.initServiceMemory`	The amount of initServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.updateServiceMemory`	The amount of updateServiceMemory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.requests.commonHooksMemory`	The amount of hook service memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	**Default Value:** 1Gi
`resources.target.averageCpuUtil`	Target CPU utilization after which Horizontal Pod Autoscaler will be triggered.	**Default Value:** 60
`minAvailable`	Minimum available pods	**Default Value:** 2
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	**Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	**Default Value:** 5
`headerIndexing.doNotIndex`	The list of headers which do not require indexing.	
`requestTimeOut`	Specifies the response time for the server to wait before timeout. This value should be updated based on the network latency.	**Default Value:** 1500 ms
`metricPrefix`	If this value is configured then the value is added as a prefix to all the metrics in the egress gateway.	
`metricSuffix`	If this value is configured then the value is added as a suffix to all the metrics in the egress gateway.	
`jaegerTelemetryTracingEnabled`	Specifies whether to enable or disable Jaeger Tracing at Egress Gateway.	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values. **Default Value:** false
`openTelemetry.jaeger.probabilisticSampler`	Specifies the Jaeger message sampler	**Default Value:** 0.5

Table 5-20 (Cont.) External Egress Gateway (`nef-egress-gateway`) Parameters

Parameter	Description	Details
`openTelemetry.jaeger.httpExporter.host`	Specifies the host of Jaeger collector service	**Default Value:** jaeger-collector.cne-infra
`openTelemetry.jaeger.httpExporter.port`	Specifies the port of Jaeger collector service	**Default Value:** 4318
`tolerations`		
`helmBasedConfigurationNodeSelectorApiVersion`	REST API version for helm based configurations	**Default Value:** v1
`nodeSelector.nodeKey`		
`nodeSelector.nodeValue`		

5.2.1.13 Diameter Gateway Parameters

Table 5-21 Diameter Gateway Parameters

Parameter	Description	Notes
`image.name`	Diameter Gateway image name	This is an optional parameter. **Default Value:** oc_nef_quality_of_service
`image.tag`	Diameter Gateway image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** Always
`replicas`	Minimum and maximum replicas to scale to maintain an average CPU utilization.	This is an optional parameter. **Default Value:** 1
`resources.limits.ephemeralStorage`	Indicates the minimum limit of ephemeral-storage.	
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1

Table 5-21 (Cont.) Diameter Gateway Parameters

Parameter	Description	Notes
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi
`envDiameterRealm`	Diameter Realm of Gateway provided in CE messages.	If no value is set, then the statefulset name is considered as default.
`envDiameterIdentity`	It is the diameter identity of gateway provided in CE messages.	If no value is set, then the pod name in statefulset is considered as default.
`envDiameterHostIp`	Contains all the k8s cluster worker node names and corresponding IP addresses.	DIAMETER_HostIp configuration must be like: node1=ip1,node2=ip2.
`staticIpAddress`	Static IP address assigned to the Load Balancer from the external load balancer IP pool.	
`staticDiamNodePort`	Specifies if static node port needs to be set.	
`service.type`	The type of the service.	
`service.port`	The port of the service.	
`service.customExtension.labels`	Custom labels that needs to be added to API Router service.	
`service.customExtension.annotations`	Custom annotations that needs to be added to API Router service.	
`deployment.customExtension.labels`	Custom lables that needs to be added to API Router deployment.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to API Router deployment.	
`log.level.diam`	Log level for Diameter logs	**Default Value:** WARN
`log.level.root`	Log level for root logs	**Default Value:** WARN
`log.level.nef`	Log level for NEF service logs	**Default Value:** WARN
`jetty.serviceMeshCheck`	Load balancing will be handled by Ingress gateway, if true it would be handled by serviceMesh	
`jetty.connectionTimeout`	Jetty client connection timeout settings	
`jetty.requestTimeout`	Jetty client request timeout settings	
`clientPeers`	Configure list of diameter client peers that can connect to diameter Gateway.	The diameter identity and realm will be validated from incoming CER request against the configuration.
`peerNodes`	Configure list of diameter peer nodes that NEF can connect.	
`peerNodes.type`	To connect to PCRF, this parameter should be set to `pcrf`. To connect to SMS-SC, this parameter should be set as `smssc`.	
`peerNodes.name`	PCRF diameter entity name.	
`peerNodes.type`	PCRF diameter entity type.	

Table 5-21 (Cont.) Diameter Gateway Parameters

Parameter	Description	Notes
peerNodes.responseOnly	PCRF diameter entity response.	
peerNodes.host	PCRF diameter entity host name.	
peerNodes.port	PCRF diameter entity port.	
peerNodes.realm	PCRF diameter entity realm.	
peerNodes.identity	PCRF diameter entity identity.	

5.2.1.14 Device Trigger Parameters

Table 5-22 Device Trigger Parameters

Parameter	Description	Notes
image.name	Device Trigger image name	This is an optional parameter. **Default Value:** oc_nef_device_trigger
image.tag	Device Trigger image tag	This is an optional parameter. **Default Value:** NEF Images
image.pullPolicy	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** Always
httpTwoEnabled		**Default Value:** true
resources.limits.cpu	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
resources.limits.hooksCpu	Maximum amount of hooks CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 0.5
resources.limits.memory	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi
resources.limits.hooksMemory	Maximum amount of hooks memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 0.5Gi
resources.requests.cpu	The amount of CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
resources.requests.hooksCpu	The amount of hooks CPU that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 0.5
resources.requests.memory	The amount of memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi

Table 5-22 (Cont.) Device Trigger Parameters

Parameter	Description	Notes
resources.requests.hooksMemory	The amount of hooks memory that the system guarantees for the resource, and Kubernetes will use this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 0.5Gi
resources.target.averageCpuUtil		
minReplicas	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 1
maxReplicas	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 1
maxUnavailable		This is an optional parameter. **Default Value:** 0
hookRestartPolicy		**Default Value:** Never
podRestartPolicy		**Default Value:** Always
nodeSelector.nodeKey		
nodeSelector.nodeValue		
extraContainers		**Default Value:** USE_GLOBAL_VALUE
terminationGracePeriodSeconds		**Default Value:** 45
log.level.root	Log level for root logs	v
log.level.nef	Log level for NEF service logs	**Default Value:** INFO
log.level.hook		**Default Value:** INFO
jetty.serviceMeshCheck	Load balancing will be handled by Ingress gateway, if true it would be handled by serviceMesh	
jetty.connectionTimeout	Jetty client connection timeout settings	**Default Value:** 10000
jetty.requestTimeout	Jetty client request timeout settings	**Default Value:** 12000
readinessProbe.initialDelaySeconds	Informs the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
readinessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10
readinessProbe.timeoutSeconds	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 3
readinessProbe.successThreshold	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
readinessProbe.failureThreshold	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 3
livenessProbe.initialDelaySeconds	tells the kubelet that it should wait xx second before performing the first probe	This is an optional parameter. **Default Value:** 25
livenessProbe.periodSeconds	specifies that the kubelet should perform a liveness probe every xx seconds	This is an optional parameter. **Default Value:** 10

Table 5-22 (Cont.) Device Trigger Parameters

Parameter	Description	Notes
`livenessProbe.timeoutSeconds`	Number of seconds after which the probe times out	This is an optional parameter. **Default Value:** 5
`livenessProbe.successThreshold`	Minimum consecutive successes for the probe to be considered successful after having failed	This is an optional parameter. **Default Value:** 1
`livenessProbe.failureThreshold`	When a Pod starts and the probe fails, Kubernetes will try failureThreshold times before giving up	This is an optional parameter. **Default Value:** 5
`service.type`	Type of the service.	This is an optional parameter. **Default Value:** ClusterIP
`service.customExtension.labels`		
`service.customExtension.annotations`		
`deployment.customExtension.labels`	Custom lables that needs to be added to API Manager service.	
`deployment.customExtension.annotations`	Custom annotations that needs to be added to API Manager service.	

5.2.1.15 MSISDNless MO SMS Parameters

Table 5-23 MSISDNless MO SMS Parameters

Parameter	Description	Notes
`image.name`	MSISDNless MO SMS image name	This is an optional parameter. **Default Value:** oc_nef_msisdnless_mo_sms
`image.tag`	MSISDNless MO SMS image tag	This is an optional parameter. **Default Value:** NEF Images
`image.pullPolicy`	Indicates if the image need to be pulled	Possible Values are: • Always • IfNotPresent • Never This is an optional parameter. **Default Value:** IfNotPresent
`shortcode_to_af_notification_destination_map.shortcode`	Indicates AF shortcode configuration.	
`shortcode_to_af_notification_destination_map.notification_url`	Indicates AF notification URL configuration.	
`jetty.serviceMeshCheck`	Load balancing is handled by Ingress gateway, if true it would be handled by serviceMesh.	
`jetty.connectionTimeout`	Jetty client connection timeout settings.	**Default Value:** 10000
`jetty.requestTimeout`	Jetty client request timeout settings.	**Default Value:** 10000

Table 5-23 (Cont.) MSISDNless MO SMS Parameters

Parameter	Description	Notes
`log.level.root`	Log level for root logs.	**Default Value:** WARN
`log.level.nef`	Log level for NEF service logs.	**Default Value:** WARN
`resources.limits.cpu`	Maximum amount of CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4
`resources.limits.hooksCpu`	Maximum amount of hooks CPU that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1
`resources.limits.memory`	Maximum amount of memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 1Gi
`resources.limits.hooksMemory`	Maximum amount of hooks memory that Kubernetes allows the job resource to use.	This is an optional parameter. **Default Value:** 4Gi
`resources.requests.cpu`	The amount of CPU that the system guarantees for the resource, and Kubernetes uses this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4
`resources.requests.hooksCpu`	The amount of hooks CPU that the system guarantees for the resource, and Kubernetes uses this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1
`resources.requests.memory`	The amount of memory that the system guarantees for the resource, and Kubernetes uses this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 4Gi
`resources.requests.hooksMemory`	The amount of hooks memory that the system guarantees for the resource, and Kubernetes uses this value to decide on which node to place the pod.	This is an optional parameter. **Default Value:** 1Gi
`resource.target.averageCpuUtil`	Average CPU Util for HPA.	
`minReplicas`	Minimum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 2
`maxReplicas`	Maximum replicas to scale to maintain an average CPU utilization	This is an optional parameter. **Default Value:** 12
`maxUnavailable`	Maximum number of PODs that can be unavailable during rolling update.	This is an optional parameter. **Default Value:** 25%
`maxSurge`	Maximum number of PODs the deployment is allowed to create at a time.	This is an optional parameter. **Default Value:** 25%

5.5 Uninstalling NEF

This section provides information about uninstalling Oracle Communications Cloud Native Core, Network Exposure Function (NEF).

> **Note**
>
> When you uninstall a Helm chart from the NEF deployment, it removes only the Kubernetes objects created during the installation.

5.5.1 Uninstalling NEF Using Helm

To uninstall NEF, run the following command:

Run the following command to completely delete or remove the NEF deployment:

```
helm uninstall <helm-release> -n <namespace>
```

Where,

helm-release is a name provided to identify the helm deployment.

namespace is the name provided to identify the namespace of NEF deployment.

For example:

```
helm uninstall ocnef -n ocnef
```

Helm keeps a record of its releases, so you can still reactivate the release after uninstalling it.

To completely remove a release from the cluster, add the `--purge` parameter to `helm delete` command:

```
helm delete --purge release_name
```

For example:

```
helm delete --purge ocnef
```

To verify the NEF uninstallation, run the following command:

```
$ kubectl get all -n <release-namespace>
```

To check the uninstallation status of the pods, use the following command:

```
kubectl get pods -<release-namespace>
```

> **Note**
>
> - During helm uninstallation if the Kubernetes jobs started by the helm hooks get stuck, then you can use the following command to delete the jobs manually:
>
>   ```
>   while true; do kubectl delete jobs --all -n <release-namespace>;
>   sleep 5;done
>   ```
>
> - If the command output displays the NEF resources or objects, then perform Deleting Kubernetes Resources.
> - If the command output displays any Kubernetes namespace, then perform the Deleting Kubernetes Namespace.

5.5.2 Deleting Kubernetes Namespace

This section describes how to delete Kubernetes namespace where NEF is deployed.

Run the following command to delete the Kubernetes namespace:

> **Note**
>
> Be sure before removing the namespace as it deletes all the resources or objects created in the namespace.

```
kubectl delete namespace <ocnef Kubernetes namespace>
```

For example:

```
kubectl delete namespace ocnef
```

To verify the NEF uninstallation, run the following command:

```
$ kubectl get all -n <release-namespace>
```

In case of successful uninstallation, no NEF resource is displayed in the command output.

If the command output displays the NEF resources or objects, then perform the following procedure:

1. Run the following command to delete all the objects:
   - To delete all the Kubernetes objects:

     ```
     kubectl delete all --all -n <release-namespace>
     ```

   - To delete all the configmaps:

     ```
     kubectl delete cm --all -n <release-namespace>
     ```

> **⚠ Caution**
>
> The command deletes all the Kubernetes objects of the specified namespace. In case you have created the RBAC resources and service accounts before the helm installation in the same namespace, and these resources are required, then do not delete them.

2. Run the following command to delete the specific resources:

   ```
   kubectl delete <resource-type> <resource-name> -n <release-namespace>
   ```

3. Run the following command to delete the Kubernetes namespace:

   ```
   kubectl delete namespace <release-namespace>
   ```

5.5.3 Deleting Kubernetes Resources

This section describes how to delete Kubernetes resources or objects where NEF is deployed.

> **ⓘ Note**
>
> Be sure before running the following commands as it deletes all the Kubernetes objects in the specified namespace.

1. Run the following command to delete all the Kubernetes objects:

   ```
   kubectl delete all --all -n <release-namespace>
   ```

2. Run the following command to delete all the configmaps:

   ```
   kubectl delete cm --all -n <release-namespace>
   ```

> **ⓘ Note**
>
> If the user has created RBAC and service account details before Helm, installing in the same namespace is required, do not delete RBAC and service account details. In case a custom service account is not provided by the user, and it is safe to remove it, then run the following commands to delete the resources or objects which are not required.

3. Run the following command to delete the specific resources:

   ```
   kubectl delete <resource-type> <resource-name> -n <release-namespace>
   ```

5.5.4 Deleting the MySQL Details

Procedure for complete removal of MySQL database and username

This procedure explains the steps to complete removal of MySQL database and users.

1. Log in to the machine which has permission to access the SQL nodes of NDB cluster.
2. Connect to the SQL node of NDB cluster successively.
3. Log in to the MySQL prompt using root permission or user, which has permission to drop the tables. For example:

   ```
   mysql -h 127.0.0.1 -uroot -p
   ```

   > **Note**
   >
   > This command may vary from system to system, path for MySQL binary, root user and root password. After running this command, the user must enter the password specific to the user mentioned in the command.

4. Run the following command to remove NEF databases:

   ```
   $ DROP DATABASE if exists <NEF database>;
   ```

   For example, to remove a database named **ocnef_db**, run the following command:

   ```
   DROP DATABASE IF EXISTS ocnef_db;
   ```

5. Run the following command to remove the NEF MySQL Users:
   Remove NEF privileged user:

   ```
   $ DROP USER IF EXISTS <NEF Privileged-User Name>;
   ```

   For example:

   ```
   $ DROP USER IF EXISTS 'priviledgeduser@'%';
   ```

   Remove NEF application user:

   ```
   $ DROP USER IF EXISTS <NEF Application User Name>;
   ```

   For example:

   ```
   $ DROP USER IF EXISTS 'appuser@'%';
   ```

   > **Note**
   >
   > Removal of MySQL Users must be done on all the SQL nodes from all the NEF sites.

6. Exit from MySQL prompt and SQL node.

6
Fault Recovery

This chapter describes the procedures to perform fault recovery for Oracle Communications Cloud Native Core, Network Exposure Function deployment.

6.2 Impacted Areas

The following table provides information about the impacted areas during NEF fault recovery:

Table 6-1 Impacted Areas

Scenario	Requires Fault Recovery or Re-install of CNE	Requires Fault Recovery or Re-install of cnDBTier	Requires Fault Recovery or Re-install of NEF
Site Failure	Yes	Yes	Yes
Migration to New Cluster	Yes, if cluster is not present.	Yes, if cnDBTier is not present.	Yes
Database Corruption	Yes, in case complete site is down.	Yes, in case replication is not up.	Yes
Deployment Corruption	No	No	Yes

6.3 Prerequisites

Before performing any fault recovery procedure, ensure that the following prerequisites are met:

- Perform the following steps to ensure that cnDBTier is in a healthy state:
  1. Run the following command to check if all the cnDBTier nodes are connected:

     ```
     ndb_mgm> show
     ```

     Example:

     ```
     ndb_mgm> show Connected to Management Server at: localhost:1186
     ```

     The output ensures that the nodes are connected.
  2. Run the following command to check the pod status:

     ```
     kubectl get pods -n <namespace>
     ```

     All the pods must be in `Running` state.
  3. Run the following command to check if the DB replication is up and running:

     ```
     mysql> show replica status\G
     ```

The values for the **Replica_IO_Running** and **Replica_SQL_Running** fields must be "Yes".

In case of any errors, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

- Enable the automatic backup on cnDBTier by scheduling regular backups. The regular backups are help in fault recovery with the following tasks:
  - Restore stable version of the network function database
  - Minimize significant loss of data due to upgrades or roll back failures
  - Minimize loss of data due to system failure
  - Minimize loss of data due to data corruption or deletion due to external input
  - Migrate network function database information from one site to another
- Ensure that the following NF specific files are available for fault recovery:
  - Custom values file (`ocnef-custom-values-<release_number>`)
  - Helm charts (`ocnef-<release_number>.tgz`)
  - Secrets and Certificates
  - RBAC resources

6.1 Overview

You must take database backup and restore it either on the same or a different cluster. It uses the NEF database to run any command or follow instructions.

> **Note**
>
> This chapter describes the scenario-based procedures to restore NEF database only. To restore all the databases that are part of cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* available on My Oracle Support.

Database Model of NEF

NEF database consists of configuration data and state data. The details are as follows:

- **Configuration Data:** The configuration data is exclusive for a given site. Thus, an exclusive logical database is created and used by the site to store its configuration data. An operator can perform configuration operations of NEF data in respective cnDBTier using the Configuration Management service.
- **State Data:** The state data is the NEF application data, which is required for service operations.

NEF has the following databases created:

- **NEF application database:** This database consists of tables used by the application to perform the functionality of the NEF network function.
- **NEF network database:** This database consists of tables used by NEF to store the network details like system details and database backups.
- **NEF common configuration database:** This database consists of tables used by common configuration.

Performing cnDBTier Fault Recovery procedures to restore cnDBTier (through automated backup file or on-demand backup from mate site), brings back configuration along with the state data captured during backup. Thus policies and other configuration will get restored, along with state data.

> **Note**
>
> If an older backup file is used to restore a site, then the stale sessions cleanup is performed automatically.

6.4 Fault Recovery Scenarios

This section describes the fault recovery procedures for various scenarios.

6.4.1 Site Failure

This is a scenario where you have cnDBTier and NEF installed on single site with automatic data replication and backup enabled. It is observed that the site has failed and there is a requirement to perform fault recovery.

Following are the assumptions:

- All the prerequisites mentioned in the Prerequisites are met.

To recover the NEF site:

1. Uninstall NEF. For information about uninstalling NEF, see the Uninstalling NEF section.
2. (Optional) Install Oracle Communications Cloud Native Environment. For information about installing CNE, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide* .
3. (Optional) Install cnDBTier. For information about installing cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
4. Perform cnDBTier fault recovery procedure by using the auto-data backup file. For more information about cnDBTier restore, see the "Restore Database with Backup" procedure in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

   > **Note**
   >
   > The auto-data backup file is the one that is built from scheduled automatic backups. The cnDBTier Backup Manager service ensures auto-data backup as per the predefined configuration. For more information about cnDBTier Backup Manager Service, see *Oracle Communications Cloud Native Core, cnDBTier User Guide*.

5. Install NEF using the helm charts. For more information about installing NEF, see the Installing NEF section.

> **⚠ Caution**
>
> This scenario may cause loss of data.

6.4.2 NEF Migration to New Cluster

This is a scenario where you want to migrate NEF to a new cnDBTier cluster. The following resons can cause the migration:

- cnDBTier is in healthy state, only NEF is migrated to a new cluster
- Database failure
- Moving NEF to a bigger cnDBTier
- Moving NEF to a new (freshly installed) cnDBTier where upgrade is not supported

> **ⓘ Note**
>
> This scenario is applicable when cnDBTier is in healthy state and NEF must be migrated to a new cluster.

Assumptions

- One of the cnDBTier is in healthy state.
- All the prerequisites mentioned in the Prerequisites are met.

To migrate NEF:

1. Take a backup of the `oc-nef-24.2.2-custom-values.yaml` file for NEF.
2. (Optional) Install CNE cluster on new site. For information about installing CNE cluster, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide* .
3. Install NEF using the `oc-nef-24.2.2-custom-values.yaml` file.

> **ⓘ Note**
>
> NEF does not provide export and import of the configuration data. You must run the configuration APIs on the freshly installed site to reconfigure the components and services.

6.4.3 Recovering Corrupted Configuration Database

This is a scenario where NEF configuration database has to be recovered from a corrupted database.

Assumptions

- One of the cnDBTier is in healthy state.
- All the prerequisites mentioned in the Prerequisites are met.

To migrate NEF:

1. Take a backup of the `oc-nef-24.2.2-custom-values.yaml` file for NEF.
2. (Optional) Install CNE cluster on new site. For information about installing CNE cluster, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide* .
3. Install NEF using the `oc-nef-24.2.2-custom-values.yaml` file.

> **Note**
>
> NEF does not provide export and import of the configuration database. You must run the configuration APIs on the freshly installed site to reconfigure the components and services.

6.4.4 Recovering NEF from Deployment Failure

This scenario describes how to recover NEF when its deployment corrupts.

Following are the assumptions:

- One of the cnDBTier is in healthy state.
- All the prerequisites mentioned in the Prerequisites are met.

To recover NEF:

1. Uninstall NEF. For information about uninstalling NEF, see the "Uninstalling NEF" section in *Oracle Communications Cloud Native Core, Network Exposure Function Installation Guide*.
2. Install NEF using the helm charts. For more information about installing NEF, see the Installing NEF section.

6.4.5 Site Failure

This section describes how to perform fault recovery when either one or all of the sites have software failure. The following are the scenarios for site failure:

Single Site Failure

A scenario where you have cnDBTier and NEF installed on two sites with automatic data replication and backup enabled. It is observed that one of the sites have failed and there is a requirement to perform fault recovery.

Following are the assumptions:

- One of the cnDBTier is in healthy state.
- All the prerequisites mentioned in the Prerequisites are met.

To recover the failed site:

1. Uninstall NEF from the failed site. For information about uninstalling NEF, see the "Uninstalling NEF" section in *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.
2. Uninstall CAPIF from the failed site. For information about uninstalling CAPIF, see the "Uninstalling CAPIF" section in *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide*.

3. (Optional) Install CNE, in case the complete site is down. For information about installing CNE, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide* .
4. (Optional) Install cnDBTier, in case replication is down or cnDBTier pods are not up and running. For information about installing cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
5. Perform cnDBTier fault recovery procedure:
    a. Perform cnDBTier fault recovery procedure to take backup from older healthy site by following the "Create On-demand Database Backup" procedure in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
    b. Restore the database to new site by following the "Restore Database with Backup" procedure in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* .
6. Install CAPIF and NEF instances using the respective custom values yaml files.

Complete Site Outage

This is a scenario where you have cnDBTier and NEF installed on two sites with automatic data replication and backup enabled, but both the sites have failed, and there is a requirement to perform fault recovery.

Following are the assumptions:

- One of the cnDBTier is in healthy state.
- All the prerequisites mentioned in the Prerequisites are met.

To recover both the sites:

1. Uninstall NEF. For information about uninstalling NEF, see the "Uninstalling NEF" section in *Oracle Communications Cloud Native Core, Network Exposure Function Installation, Upgrade, and Fault Recovery Guide* .
2. Uninstall CAPIF. For information about uninstalling CAPIF, see the Uninstalling CAPIF section.
3. (Optional) Install Oracle Communications Cloud Native Environment. For information about installing CNE, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.
4. (Optional) Install cnDBTier. For information about installing cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* .
5. Perform cnDBTier fault recovery procedure.Use auto-data backup file for restore procedure. For more information about cnDBTier restore, see the "Restore Database with Backup" procedure in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide* .

> **Note**
>
> The auto-data backup file is one that built from scheduled automatic backup. cnDBTier Backup Manager Service ensures auto-data backup as per the predefined configuration. For more information about cnDBTier Backup Manager Service, see *Oracle Communications Cloud Native Core, cnDBTier User Guide*.

6. Install NEF using the helm charts on all the sites with site-specific custom values. For more information about installing NEF, see the Installing NEF section.

> **⚠ Caution**
>
> There might be a data loss in this scenario.

A

PodDisruptionBudget Kubernetes Resource

PodDisruptionBudget (PDB) is a Kubernetes resource that allows you to achieve High Availability (HA) of scalable application services when the cluster administrators perform voluntary disruptions to manage the cluster nodes.

PDB restricts the number of simultaneously down pods due to voluntary disruptions. It is helpful in running services undisrupted in case, a pod is deleted accidentally or deliberately.

PDB is defined for HA, scalable NEF and scalable CAPIF services such as apirouter, monitoring event, ccf client, fivegcagent, qualityofservice, expiry-auditor, api manager, af manager, event manager, ocegress_gateway, and ocingress_gateway.

It allows safe eviction of pods when a Kubernetes node is drained to perform maintenance on the node. It uses the default value set for the `maxUnavailable` parameter in the Helm chart to determine the maximum number of pods that can be unavailable during a voluntary disruption.

For more information about this functionality, see https://kubernetes.io/docs/concepts/workloads/pods/disruptions/#pod-disruption-budgets.

> **Note**
>
> The performance and capacity of the NEF system may vary based on the call model, feature or interface configuration, and underlying CNE and hardware environment.

The following table describes the predefined PDB values configured for each service in the NEF deployment:

Table A-1 Predefined PDB values NEF deployment

Microservice	Specified Number/Percentage of Pods for PDB (maxUnavailable)
ccfclient manager	0%
aef_apirouter	0%
monitoringevents	0%
fivegcagent	0%
apd_manager	0%
expiry-auditor	0%
qualityofservice	0%
device_trigger	0%
diam_gateway	0%
pool_manager	0%
traffic_influence	0%
msisdnless_mo_sms	0%
console_data_service	25%
egress-gateway	1

Table A-1 (Cont.) Predefined PDB values NEF deployment

Microservice	Specified Number/Percentage of Pods for PDB (maxUnavailable)
ingress-gateway	1

The following table describes the predefined PDB values configured for each service in the CAPIF deployment:

Table A-2 Predefined PDB values for CAPIF deployment

Microservice	Specified Number/Percentage of Pods for PDB (maxUnavailable)
afmgr	0%
apimgr	0%
eventmgr	0%
egress-gateway	1
ingress-gateway	1

B

Adding a Site to an Existing Deployment

This section describes the procedure to add a site to an existing Oracle Communications Cloud Native Core, Network Exposure Function (NEF) site.

Prerequisites

- NEF connected to a cnDBTier is up and running. This is referred as Site-1 (CAPIF-1 and NEF-1).
- The CAPIF, NEF, and cnDBTier versions used for Site-1 installation must be identified and same must be used for adding another site.

Adding a Site

1. Install a new cnDBTier. This cnDBTier must act as a georedundant database to the cnDBTier in Site-1. For more information to install a cnDBTier, see *Oracle Communications Cloud Native Core, cnDBTier User Guide*.
2. Verify the replication channel between the cnDBTier sites by sending the following request to the dbMonitor service of both the cnDBTier sites. The responses from both the cnDBTier sites must show the status of replication channel as up:

   ```
   curl http://<mysql-db-monitor-service>:8080/db-tier/status/replication/
   realtime
   ```

   Sample command:

   ```
   curl http://mysql-cluster-db-monitor-svc:8080/db-tier/status/replication/
   realtime
   ```

   Sample output:

   ```
   [{"localSiteName":"Site-1","remoteSiteName":"Site-2","replicationStatus":"U
   P","secondsBehindRemote": 0}]
   ```

3. Install CAPIF on Site-2. For more information on installation procedure, see Installing CAPIF section.
4. Perform the upgrade for NEF on Site-1 to add the details of CAPIF-2 in the `ocnef-custom-values.yaml` file.
   Sample configuration:

   ```
   publicKeyMonitorDelay: 15000
   capifDetails:
           - capifInstanceId: f98b05e0-22d8-11ed-861d-0242ac120002
             type: local # this parameter would be used to populate the
   capif_instance_id column for existing records in upgrade to 22.4.0
             host: 10.122.123.123
             port: 8080
             scheme: http
   ```

```
    certificate:
      secretName: certificate-secret
      certificateName: tmp.cer


  - capifInstanceId: f98b05e0-22d8-11ed-861d-0242ac120003
    type: remote
    host: 10.122.123.223
    port: 8080
    scheme: http
    certificate:
      secretName: certificate-secret
      certificateName: tmp.cer
```

5. Install NEF on Site-2 by providing the details about both the CAPIF instances in the `ocnef-custom-values.yaml`. For more information on installation procedure, see Installing NEF section.

C

Removing a Site to from an Existing Georedundant Deployment

Prerequisites

- NEF connected to a cnDBTier is up and running. This is referred as Site-1 (CAPIF-1 and NEF-1).

This section describes the procedure to remove a site (Site-2) from an existing georedundant NEF deployment.

1. Offboard all the API invokers from CAPIF of the site that needs to be removed.
   This deletes all the subscriptions related to the API invokers onboarded on the CAPIF that needs to be removed.
2. Upgrade the NEF instance on the surviving site to remove the details of the CAPIF of the site to be removed.
3. Uninstall the CAPIF instance from the site to be removed.

> **Note**
>
> - The CAPIF instance removed from a georedundant deployment must not be used in any other NEF deployments or as an standalone CAPIF. It is recommended to uninstall the CAPIF and then install again.
> - NEF supports only 2-site.

4. Uninstall the NEF instance from the site to be removed.
5. Remove the DB Replication and the NDBCluster on the site to be removed.

Example:
The following steps provide an example to remove Site-2 that consists of NEF-2 and CAPIF-2 whereas, Site-1 with NEF-1 and CAPIF-1 is the surviving site:

1. Offboard all the invokers from CAPIF-2 on Site-2.
2. Upgrade NEF-1 to remove details of CAPIF-2 from the `ocnef-custom-values.yaml` file during upgrade on Site-1.
3. Uninstall CAPIF-2 instance from Site-2.
4. Uninstall NEF-2 from Site-2.
5. Remove the DB Replication and the NDBCluster on Site-2.