Oracle® Communications

Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide

Release 24.3.0
G13506-01
November 2024

Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide, Release 24.3.0

G13506-01

Contents

3 Customizing Policy

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

Acronym	Description
3GPP	3rd Generation Partnership Project
AAA	Authorization Authentication Answer
AAR	Authorization Authentication Request
AF	Application Function
AMF	Access and Mobility Management Function
API	Application Programming Interface
ARS	Alternate Route Selection
ASM	Aspen Service Mesh
ASR	Abort-Session-Request
ATS	The core service sends the subscriber state variables to PDS only when there is an update to the variables.
AVP	Attribute Value Pair
BSF	Oracle Communications Cloud Native Core, Binding Support Function
CA	Certificate Authority
CDCS	Oracle Communications CD Control Server
CHF	Charging Function
CM	Configuration Management
CNC	Cloud Native Core
CNC Console	Oracle Communications Cloud Native Configuration Console
CNE	Oracle Communication Cloud Native Core, Cloud Native Environment
CNLB	Cloud Native Load Balancer
CNPCRF	Oracle Communications Cloud Native Core, Policy and Charging Rules Function
CUSTOMER_REPO	Docker registry address including the port number, if the docker registry has an associated port.
cnDBTier	Oracle Communications Cloud Native Core, cnDBTier
DNS	Domain Name System
DRA	Diameter Routing Agent
FQDN	Fully Qualified Domain Name
GUAMI	Globally Unique AMF Identifier
IMAGE_TAG	Image tag from release tar file. You can use any tag number. However, make sure that you use that specific tag number while pushing docker image to the docker registry.
IMS	IP Multimedia Subsystem
HTTPS	Hypertext Transfer Protocol Secure
MCC	Mobile Country Code
MCPTT	Mission-critical push-to-talk
METALLB_ADDRESS_POOL	Address pool configured on metallb to provide external IPs
MNC	Mobile Network Code

Table (Cont.) Acronyms

Acronym	Description
NAD	Network Attachment Definitions
NEF	Oracle Communications Cloud Native Core, Network Exposure Function
NF	Network Function
NPLI	Network Provided Location Information
NRF	Oracle Communications Cloud Native Core, Network Repository Function
OSO	Oracle Communications Operations Services Overlay
P-CSCF	Proxy Call Session Control Function
PA Service	Policy Authorization Service
PCC	Policy and Charging Control
PDB	Pod Disruption Budget
PLMN	Public Land Mobile Network
PCF	Oracle Communications Cloud Native Core, Policy Control Function
PCRF	Oracle Communications Cloud Native Core, Policy and Charging Rules Function
PCEF	Policy and Charging Enforcement Function
PCSCF	Proxy Call Session Control Function
PDS	Policy Data Service
PRA	Presence Reporting Area
PRE	Policy Runtime Engine
PDU	Protocol Data Unit
Policy	Oracle Communications Cloud Native Core, Converged Policy
QoS	Quality of Service
RAA	Re-Auth-Answer
RAN	Radio Access Network
RAR	Re-Auth-Request
SBI	Service Based Interface
SAN	Subject Alternate Name
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
UDR	Oracle Communications Cloud Native Core, Unified Data Repository
SRA	Successful Resource Allocation
STR	Session Termination Request
TTL	Time To Live
UE	User Equipment
UPF	User Plane Function
UPSI	UE Policy Section Identifier
URSP	UE Route Selection Policies
UPSC	UE Policy Section Code
URI	Uniform Resource Identifier
VSA	Vendor Specific Attributes

What's New in This Guide

This section introduces the documentation updates for release 24.3.x.

Release 24.3.0 - G13506-01, November 2024

Generic Updates:

- Added the following parameters to Diameter Gateway and Diameter Connector Configuration to configure Diameter Gateway for handling stale requests cleanup for PCRF Core:
  - `ENABLE_LATE_ARRIVAL`
  - `ENABLE_LATE_PROCESSING`
  - `LATE_ARRIVAL_MAX_RESPONSE_TIME`
  - `SKIP_LATE_PROCESSING_FOR_TERMINATE`
- Added `ENABLE_LATE_ARRIVAL` to PCRF-Core Configurations.
- Added the TLS Configurations section to include details of configurations required for TLS.
- Added Creating Secret for Support of TLS in Diameter Gateway to describe the step to create secret for TLS in Daimeter Gateway.
- Added the following parameters to TLS Configuration in Diameter Gateway:
  - `TLS_ENABLED`
  - `TLS_DIAMETER_PORT`
  - `TLS_CIPHER_SUITE`
  - `TLS_INITIAL_ALGORITHM`
  - `TLS_SECRET_NAME`
  - `TLS_RSA_PRIVATE_KEY_FILENAME`
  - `TLS_ECDSA_PRIVATE_KEY_FILENAME`
  - `TLS_RSA_CERTIFICATE_FILENAME`
  - `TLS_ECDSA_CERTIFICATE_FILENAME`
  - `TLS_CA_BUNDLE_FILENAME`
  - `TLS_MTLS_ENABLED`
- Updated the API Root Configuration for Resource URI and Notification URI section with information related to traffic segregation using CNLB.
- Added the `global.swaggerUiEnabled` parameter in Enabling/Disabling Services Configurations section to enable Swagger Console for UDR Connector and CHF Connector Services.
- Added the `bulwark.congestion.responseCode` parameter in Bulwark Service Configuration section to configure the response code for rejected requests during Bulwark Pod congestion.

Installation Updates:

- Added preinstallation steps for Configuring Traffic Segregation.

- Updated the Pushing the Images to Customer Docker Registry section to describe the compatible docker image versions for various Policy microservices.
- Updated the Preupgrade Tasks section with the details of manually creating database slicing tables for SM service, Usage Monitoring service, and PCRF Core when upgrading Policy from a version in which the database slicing was introduced.
- Updated configuration details of Service Entries, Destination Rule, and envoyFilters in Installing Service Mesh Configuration Charts.

Upgrade and Rollback Updates:

- Updated the Supported Upgrade Paths in Supported Upgrade Paths.
- Updated the Supported RollBack Paths in Rolling Back Policy.

1 Introduction

This guide describes how to install or upgrade Oracle Communications Cloud Native Core, Converged Policy (Policy) in a cloud native environment. It also includes information on performing fault recovery for Policy.

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 their deployed container runtime environment.
- `kubectl` commands might 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 can 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

Policy is a key component of the 5G Service Based Architecture(SBA). It provides a flexible, secure, and scalable policy designing solution. Policy interacts with other network functions to perform usage monitoring, network behavior management, and governance. It helps operators to design, test, and deploy different network policies supporting 5G deployments. Policy is designed and built with a microservice based architecture on cloud native principles. It uses network, subscriber, and service information to help service providers create policies and determine how and under what conditions subscribers and applications use network resources. It helps in minimizing network utilization while maximizing the quality of experience for operators. Policy solution supports deployments into any cloud, including containers on Bare Metal managed by Kubernetes or VMs managed by OpenStack.

Note

The performance and capacity of the Policy system may vary based on the Call model, Feature/Interface configuration, underlying CNE and hardware environment, including but not limited to the complexity of deployed policies, policy table size , object expression and custom json usage in Policy design.

Policy is a network function for policy control decision and flow based charging control. It consists of the following functions:

- Policy rules for application and service data flow detection, gating, QoS, and flow based charging to the Session Management Function (SMF)
- Access and Mobility Management related policies to the Access and Mobility Management Function (AMF)
- UE Route Selection Policies (URSP) rules to User Equipement (UE) through AMF
- Access to subscription information relevant for policy decisions in a Unified Data Repository (UDR)
- Network control for service data flow detection, gating, and Quality of Service (QoS)
- Flow based charging towards the Policy and Charging Enforcement Function (PCEF)
- Receiving session and media related information from Application Function (AF) and informing AF of traffic plane events
- Provision of Policy and Charging Control (PCC) Rules to Policy and Charging Enforcement Function (PCEF) through the Gx reference point

Policy supports the above functions through the following services:

- Session Management Service
- Access and Mobility Service
- Policy Authorization Service
- User Equipment (UE) Policy Service
- PCRF Core Service
- Binding Service
- Policy Data Source Service
- Usage Monitoring Service
- Notifier Service
- NWDAF Agent

1.2 References

Refer to the following documents while deploying Policy:

- *Oracle Communications Cloud Native Core, Converged Policy User Guide*
- *Oracle Communications Cloud Native Core, Converged Policy REST API Specification Guide*
- *Oracle Communications Cloud Native Core, Converged Policy Design Guide*
- *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*
- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation and Upgrade Guide*
- *Oracle Communications Cloud Native Core, cnDBTier User Guide*
- *Oracle Communications Cloud Native Core, Data Collector User Guide*

1.3 Oracle Error Correction Policy

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

Table 1-1 Premier Support Details

Release Number	General Availability (GA) Date	Latest Patch Version	Error Correction Grace Period End Date
3.24.3	November 2024	24.3.0	November 2025
3.24.2	November 2024	24.2.2	November 2025
3.24.1	April 2024	24.1.0	April 2025
3.23.4	November 2024	23.4.8	November 2025

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

1.4 Oracle 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 Oracle Open Source Support Policies.

2
Installing Policy

This chapter provides information about installing Oracle Communications Cloud Native Core, Converged Policy (Policy) in a cloud native environment.

> **Note**
>
> Policy supports fresh installation, and it can also be upgraded from CNC Policy 24.2.x and 24.1.x. For more information on how to upgrade Policy, see Upgrading Policy.

Deployment Models

Policy can be deployed in different modes based on the network requirements. The following deployment models supported for Policy are as follow:

- **Converged Policy** – Unified policy solution that supports both PCF and PCRF functionalities. If the user wants to enable only PCRF service, enable PCRF and its related services, and disable the PCF services.
- **PCF only** - Independent deployment for PCF and its microservices.

2.1 Prerequisites

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

2.1.1 Software Requirements

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

Table 2-1 Preinstalled Software

Software	Versions
Kubernetes	1.30.x, 1.29.x,1.28.x
Helm	3.14.2
Podman	4.9.4

> **Note**
>
> CNE 24.3.x, 24.2.x, and 24.1.x versions can be used to install Policy 24.3.0.

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

```
kubectl version
```

```
helm version
```

```
podman version
```

```
docker version
```

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

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	Version	Purpose
AlertManager	0.27.0	Alerts Manager
Calico	3.27.3	Security Solution
cert-manager	1.12.4	Secrets Manager
Containerd	1.7.16	Container Runtime Manager
Fluentd - OpenSearch	1.16.2	Logging
Grafana	9.5.3	Metrics
HAProxy	3.0.2	Load Balancer
Istio	1.18.2	Service Mesh
Jaeger	1.60.0	Tracing
Kyverno	1.12.5	Logging
MetalLB	0.14.4	External IP
Oracle OpenSearch	2.11.0	Logging
Oracle OpenSearch Dashboard	2.11.0	Logging
Prometheus	2.52.0	Metrics
Prometheus Operator	0.76.0	Metrics
Velero	1.12.0	Logging
elastic-curator	5.5.4	Logging
elastic-exporter	1.1.0	Logging
elastic-master	7.9.3	Logging
Logs	3.1.0	Logging
prometheus-kube-state-metric	1.9.7	Metrics
prometheus-node-exporter	1.0.1	Metrics

Table 2-2 (Cont.) Additional Software

Software	Version	Purpose
metrics-server	0.3.6	Metric Server
occne-snmp-notifier	1.2.1	Metric Server
tracer	1.22.0	Tracing

Important

If you are using NRF with Policy, install it before proceeding with the Policy installation. Policy 24.3.0 supports NRF 24.3.x.

2.1.2 Environment Setup Requirements

This section describes the environment setup requirements for installing Policy.

2.1.2.1 Network Access Requirement

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

- Local Helm repository: It contains the Policy 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 Policy Docker images.
  To check if the Kubernetes cluster hosts can access the local Docker image repository, pull any image with an image-tag, using either of the following commands:

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

  ```
  podman pull <podman-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 Policy pod.

For example:

```
docker pull CUSTOMER_REPO/oc-app-info:24.3.5
```

```
podman pull occne-repo-host:5000/occnp/oc-app-info:24.3.5
```

2.1.2.2 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:

- network access to the Helm repository and Docker image repository
- Helm repository configured
- network access to the Kubernetes cluster
- required environment settings to run the `kubectl`, `podman`, and `docker` commands. The environment must have privileges to create 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

2.1.2.3 Server or Space Requirement

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

2.1.2.4 CNE Requirement

This section is applicable only if you are installing Policy on Oracle Communications Cloud Native Core, Cloud Native Environment. Policy supports CNE 24.3.x, 24.2.x, and 24.1.x

To check the CNE version, run the following command:

```
echo $OCCNE_VERSION
```

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

2.1.2.5 cnDBTier Requirement

Policy supports cnDBTier 24.3.x, 24.2.x, and 24.1.x. cnDBTier must be configured and running before installing Policy. For more information about cnDBTier installation procedure, see *Oracle Communications Cloud Native Core, cnDBTier Installation,Upgrade and Fault Recovery Guide*.

Recommended MySQL Configurations
Following are the modified or additional parameters for cnDBTier:

> **Note**
>
> These parameters must be updated in the cnDBTier custom_values.yaml before the installation and upgrade.

Table 2-3 Modified/Additional cnDBTier Parameters

Parameter	Modified/Additional	Old value	Current Value
ndb_batch_size	Modified	0.03G	2G
TimeBetweenEpochs	Modified	200	100
NoOfFragmentLogFiles	Modified	128	50
FragmentLogFileSize	Modified	16M	256M
RedoBuffer	Modified	32M	1024M
ndbmtd pods CPU	Modified	3/3	8/8
ndb_report_thresh_binlog_epoch_slip	Additional	NA	50
ndb_eventbuffer_max_alloc	Additional	NA	19G
ndb_log_update_minimal	Additional	NA	1
replicationskiperrors	Modified	enable: false	enable: true
replica_skip_errors	Modified	'1007,1008,1050,1051,1022,1296,13119'	'1007,1008, 1022, 1050,1051,1054,1060,1061,1068,1094,1146, 1296,13119'

Important

These parameters may need further tuning based on the call model and deployments. For any further support, you must consult My Oracle Support (https://support.oracle.com).

2.1.2.6 OSO Requirement

Policy supports Operations Services Overlay (OSO) 24.3.x, 24.2.x, and 24.1.x for common operation services (Prometheus and components such as alertmanager, pushgateway) on a Kubernetes cluster, which does not have these common services. For more information about OSO installation, see *Oracle Communications Cloud Native Core, Operations Services Overlay Installation, Upgrade, and Fault Recovery Guide*.

2.1.2.7 CNC Console Requirements

Policy supports CNC Console (CNCC) 24.3.x

For more information about CNCC, see *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide and Oracle Communications Cloud Native Configuration Console User Guide.*

2.1.2.8 OCCM Requirements

Policy supports OCCM 24.3.x. To support automated certificate lifecycle management, Policy integrates with Oracle Communications Cloud Native Core, Certificate Management (OCCM) in compliance with 3GPP security recommendations. For more information about OCCM in Policy, see the *Support for Automated Certificate Lifecycle Management* section in *Oracle Communications Cloud Native Core, Converged Policy User Guide*.

For more information about OCCM, see the following guides:

- *Oracle Communications Cloud Native Core, Certificate Manager Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Certificate Manager User Guide*

2.1.2.9 OCNADD Requirements

Policy supports Oracle Communications Network Analytics Data Director (OCNADD) <version> to store the metadata of the messages copied at Ingress Gateway and Egress Gateway, Storing these messages is required to support SBI monitoring.

For more information about copying the messages at Ingress and Egress gateways, see *Message Feed for SBI Monitoring* section in *Oracle Communications Cloud Native Core, Converged Policy User Guide*.

For details on Oracle Communications Network Analytics Data Director (OCNADD), see *Oracle Communications Network Analytics Data Director User Guide*.

2.1.3 Resource Requirements

This section lists the resource requirements to install and run Policy.

> **Note**
>
> The performance and capacity of the Policy system may vary based on the Call model, Feature/Interface configuration, underlying CNE and hardware environment, including but not limited to the complexity of deployed policies, policy table size, object expression and custom json usage in policy design.

2.1.3.1 Policy Services

The following table lists resource requirement for Policy Services:

Table 2-4 Policy Services:

Service Name	CPU		Memory(Gi)		Replica(s)			Ephemeral-Storage	
	Min	Max	Min	Max	Count	Min	Max	Min	Max
App-Info	2	2	4	4	1	2	5	80Mi	1Gi
Audit Service	2	2	4	4	1	2	8	80Mi	1Gi
CM Service	2	4	0.5	2	2	NA	NA	80Mi	1Gi
Config Server	4	4	0.5	2	1	1	2	80Mi	1Gi
NRF Client NF Discovery	4	4	2	2	2	2	5	80Mi	1Gi

Table 2-4 (Cont.) Policy Services:

Service Name	CPU		Memory(Gi)		Replica(s)			Ephemeral-Storage	
	Min	Max	Min	Max	Count	Min	Max	Min	Max
NRF Client NF Management	1	1	1	1	2	NA	NA	80Mi	1Gi
Perf-Info	1	2	1	2	2	NA	NA	80Mi	1Gi
PRE-Test	1	1	0.5	2	1	1	8	80Mi	1Gi
Query Service	1	2	1	1	1	1	2	80Mi	1Gi
Soap Connector	2	4	4	4	2	2	8	80Mi	1Gi
NWDAF Agent	1	2	1	1	1	1	1	80Mi	1Gi
Alternate Route Service	2	2	4	4	1	2	5	80Mi	1Gi
Binding Service	6	6	8	8	1	2	8	80Mi	1Gi
Bulwark Service	8	8	6	6	2	2	8	80Mi	1Gi
Egress Gateway	4	4	6	6	2	2	5	80Mi	1Gi
Ingress Gateway	5	5	6	6	2	2	5	80Mi	1Gi
LDAP Gateway	3	4	2	4	1	2	4	80Mi	1Gi
Policy Data Source (PDS)	7	7	8	8	1	2	8	80Mi	1Gi
PRE-Test	1	1	0.5	2	1	1	8	80Mi	1Gi
Notifier Service	1	2	1	1	2	2	8	80Mi	1Gi
Usage Monitoring	4	5	3	4	2	2	4	80Mi	1Gi
Diameter Connector	4	4	1	2	1	2	8	80Mi	1Gi
Diameter Gateway	4	4	1	2	1	2	[1]	80Mi	1Gi
PCRF-Core	8	8	8	8	2	2		80Mi	1Gi
AM Service	8	8	8	8	1	2		80Mi	1Gi
SM Service	7	7	10	10	2	2		80Mi	1Gi
UE Service	8	8	6	6	2	2		80Mi	1Gi
UDR-Connector	6	6	4	4	2	2		80Mi	1Gi

Table 2-4 (Cont.) Policy Services:

Service Name	CPU		Memory(Gi)		Replica(s)			Ephemeral-Storage	
	Min	Max	Min	Max	Count	Min	Max	Min	Max
CHF-Connector	6	6	4	4	2	2		80Mi	1Gi

1 Max replica per service should be set based on required TPS and other dimensioning factors.

Upgrade resources should be taken into account during dimensioning. Default upgrade resource requirements are 25% above maximum replica, rounding up to the next integer. For example, if a service has a max replica count of 8, upgrade resources of 25% will result in additional resources equivalent to 2 pods.

Updating CPU and Memory for Microservices

To modify the default values of replicas for any Policy microservice, you can add the following parameters under the required service group with CPU and memory in `occnp_custom_values_24.3.0.yaml` file:

```
minReplicas: 1
maxReplicas: 1
```

For example, to update the default values for Ingress gateway or Egress gateway, add the parameters under `ingress-gateway` or `egress-gateway` group:

```
ingress-gateway:
  #Resource details
  resources:
    limits:
      cpu: 1
      memory: 6Gi
    requests:
      cpu: 1
      memory: 2Gi
    target:
      averageCpuUtil: 80
  minReplicas: 1
  maxReplicas: 1
egress-gateway:
  #Resource details
  resources:
    limits:
      cpu: 1
      memory: 6Gi
    requests:
      cpu: 1
      memory: 2Gi
    target:
      averageCpuUtil: 80
  minReplicas: 1
  maxReplicas: 1
```

> **Note**
>
> It is recommended to avoid altering the above mentioned standard resources. Either increasing or decreasing the CPU or memory will result in unpredictable behavior of the pods. Contact My Oracle Support (MOS) for Min Replicas and Max Replicas count values.

By default, ephemeral storage resource is enabled during Policy deployment with request set to 80Mi and limit set to 1Gi. These values can be updated by modifying the custom-values.yaml file using the global Helm variables `logStorage` and `crictlStorage`.

To disable ephemeral storage resources, add these variables in the global section and replace their default value to 0:

- `logStorage`: 70 #default calculated value 70
- `crictlStorage`: 3 #default calculated value 1

To change the ephemeral storage limit value of a service, ensure that at least one of the global Helm variables `logStorage` and `crictlStorage` is non-zero and modify the ephemeral storage limit of the service in custom-values.yaml.

To change the ephemeral storage request value of services, ensure that at least one of the global Helm variables `logStorage` and `crictlStorage` is non-zero and 110% of their summation determines the ephemeral storage request of all the services in custom-values.yaml.

> **Note**
>
> In certain scenarios, data collection like full heap dump requires additional ephemeral storage limit values. In such cases, the ephemeral storage resources must be modified in pod's deployment.

2.1.3.2 Upgrade

The following Upgrade Resources are required for each microservice mentioned in the below table:

Table 2-5 Upgrade

Service Name	CPU Min	CPU Max	Memory Min (Gi)	Memory Max (Gi)	Ephemeral Storage Min	Ephemeral Storage Max	Replica Count
AM Service	1	2	1	2	200Mi	2Gi	1
Audit Service	1	2	1	2	200Mi	2Gi	1
Binding Service	1	2	1	2	200Mi	2Gi	1
Config Server	1	2	1	2	200Mi	2Gi	1
PCRF-Core	1	2	1	2	200Mi	2Gi	1

Table 2-5 (Cont.) Upgrade

Service Name	CPU Min	CPU Max	Memory Min (Gi)	Memory Max (Gi)	Ephemeral Storage Min	Ephemeral Storage Max	Replica Count
Policy Data Source (PDS)	1	2	1	2	200Mi	2Gi	1
SM Service	1	2	1	2	200Mi	2Gi	1
UE Service	1	2	1	2	200Mi	2Gi	1
UDR-Connector	1	2	1	2	200Mi	2Gi	1
CHF-Connector	1	2	1	2	200Mi	2Gi	1
Usage Monitoring	1	2	1	2	200Mi	2Gi	1

2.2 Installation Sequence

This section describes preinstallation, installation, and postinstallation tasks for Policy.

2.2.1 Preinstallation Tasks

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

2.2.1.1 Downloading Policy package

To download the Policy package from My Oracle Support (MOS), perform the following steps:

1. Log in to My Oracle Support with your credentials.
2. Select the **Patches and Updates** tab.
3. In the **Patch Search** window, click **Product or Family (Advanced)** option.
4. Enter *Oracle Communications Cloud Native Core - 5G* in the **Product** field, and select the Product from the drop-down list.
5. From the **Release** drop-down list, select "**Oracle Communications Cloud Native Core, Converged Policy <release_number>**".
   Where, `<release_number>` indicates the required release number of Policy.
6. Click **Search**.
   The Patch Advanced Search Results lists appears
7. Select the required patch from the results.
   The Patch Details window papers.
8. Click **Download**.
   File Download window appears.

9. Click the *<p********_<release_number>_Tekelec>.zip* file to download the Policy release package.

2.2.1.2 Pushing the Images to Customer Docker Registry

Policy deployment package includes ready-to-use images and Helm charts to orchestrate containers in Kubernetes. The communication between Pods of services of Policy products are preconfigured in the Helm charts.

Table 2-6 Docker Images for Policy

Service Name	Image Name	Image Tag
Alternate Route Service	alternate_route	24.3.3
AM Service	oc-pcf-am	24.3.0
Application Info Service	oc-app-info	24.3.5
Binding Service	oc-binding	24.3.0
Bulwark Service	oc-bulwark	24.3.0
CM Service	oc-config-mgmt	24.3.5
CM Service	common_config_hook	24.3.3
Config Server Service	oc-config-server	24.3.5
Debug Tool	ocdebug-tools	24.3.1
Diameter Connector	oc-diam-connector	24.3.5
Diameter Gateway	oc-diam-gateway	24.3.5
Egress Gateway	ocegress_gateway	24.3.3
NF Test	nf_test	24.3.2
Notifier Service	oc-notifier	24.3.0
Ingress Gateway	ocingress_gateway	24.3.3
Ingress Gateway/Egress Gateway init configuration	configurationinit	24.3.3
Ingress Gateway/Egress Gateway update configuration	configurationupdate	24.3.3
LDAP Gateway Service	oc-ldap-gateway	24.3.0
Nrf Client Service	nrf-client	24.3.2
NWDAF Agent	oc-nwdaf-agent	24.3.0
PCRF Core Service	oc-pcrf-core	24.3.0
Performance Monitoring Service	oc-perf-info	24.3.5
PolicyDS Service	oc-policy-ds	24.3.0
Policy Runtime Service	oc-pre	24.3.0
Query Service	oc-query	24.3.5
Session State Audit	oc-audit	24.3.5
SM Service	oc-pcf-sm	24.3.0
Soap Connector	oc-soap-connector	24.3.0
UE Service	oc-pcf-ue	24.3.0
Usage Monitoring	oc-usage-mon	24.3.0
User Service	oc-pcf-user	24.3.0

Pushing Images

To Push the images to customer docker resgistry, perform the following steps:

1. Unzip the release package to the location where you want to install Policy.
   `tar -xvzf occnp-pkg-24.3.0.0.0.tgz`

   The directory consists of the following:

   - `occnp-images-24.3.0.tar`: Policy Image File
   - `occnp-24.3.0.tgz` : Helm file
   - `Readme.txt`: Readme txt File
   - `occnp-24.3.0.tgz.sha256`: Checksum for Helm chart tgz file
   - `occnp-servicemesh-config-24.3.0.tgz.sha256`: Checksum for Helm chart for Service Mesh tgz file
   - `occnp-images-24.3.0.tar.sha256`: Checksum for images' tgz file
2. Run one of the following commands to load `occnp-images-24.3.0.tar` file

   ```
   docker load --input /IMAGE_PATH/occnp-images-24.3.0.tar
   ```

   ```
   podman load --input /IMAGE_PATH/occnp-images-24.3.0.tar
   ```
3. To verify if the image is loaded correctly, run one of the following commands:

   ```
   docker images
   ```

   ```
   podman images
   ```

   Verify the list of images shown in the output with the list of images shown in the table. If the list does not match, reload the image tar file.
4. Create a new tag for each imported image and push the image to the customer docker registry by entering the following commands:

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

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

   Where,

   - `<image-name>` is the image name.
   - `<image-tag>` is the image release number.
   - `<docker-repo>` is the docker registry address with Port Number if registry has port attached. This is a repository to store the images.
   - `<podman-repo>` is the Podman registry address with Port Number if registry has port attached. This is a repository to store the images.

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

Example for CNE 1.8 and later

```
podman tag docker.io/occnp/oc-app-info:24.3.5 occne-repo-host:5000/occnp/oc-
app-info:24.3.5
podman push occne-repo-host:5000/occnp/oc-app-info:24.3.5

podman tag docker.io/occnp/nf_test:24.3.2 occne-repo-host:5000/occnp/
nf_test:24.3.2
podman push occne-repo-host:5000/occnp/nf_test:24.3.2

podman tag docker.io/occnp/oc-policy-ds:24.3.0 occne-repo-host:5000/occnp/oc-
policy-ds:24.3.0
podman push occne-repo-host:5000/occnp/oc-policy-ds:24.3.0

podman tag docker.io/occnp/alternate_route:24.3.3 occne-repo-host:5000/occnp/
alternate_route:24.3.3
podman push occne-repo-host:5000/occnp/alternate_route:24.3.3

podman tag docker.io/occnp/ocingress_gateway:24.3.3 occne-repo-host:5000/
occnp/ocingress_gateway:24.3.3
podman push occne-repo-host:5000/occnp/ocingress_gateway:24.3.3

podman tag docker.io/occnp/oc-pcf-sm:24.3.0 occne-repo-host:5000/occnp/oc-pcf-
sm:24.3.0
podman push occne-repo-host:5000/occnp/oc-pcf-sm:24.3.0

podman tag docker.io/occnp/oc-pcf-am:24.3.0 occne-repo-host:5000/occnp/oc-pcf-
am:24.3.0
podman push occne-repo-host:5000/occnp/oc-pcf-am:24.3.0

podman tag docker.io/occnp/oc-pcf-ue:24.3.0 occne-repo-host:5000/occnp/oc-pcf-
ue:24.3.0
podman push occne-repo-host:5000/occnp/oc-pcf-ue:24.3.0

podman tag docker.io/occnp/oc-audit:24.3.5 occne-repo-host:5000/occnp/oc-
audit:24.3.5
podman push occne-repo-host:5000/occnp/oc-audit:24.3.5

podman tag docker.io/occnp/oc-ldap-gateway:24.3.0 occne-repo-host:5000/
occnp/oc-ldap-gateway:24.3.0
podman push occne-repo-host:5000/occnp/oc-ldap-gateway:24.3.0

podman tag docker.io/occnp/oc-query:24.3.5 occne-repo-host:5000/occnp/oc-
query:24.3.5
podman push occne-repo-host:5000/occnp/oc-query:24.3.5

podman tag docker.io/occnp/oc-pre:24.3.0 occne-repo-host:5000/occnp/oc-
pre:24.3.0
podman push occne-repo-host:5000/occnp/oc-pre:24.3.0
```

```
podman tag docker.io/occnp/oc-perf-info:24.3.5 occne-repo-host:5000/occnp/oc-
perf-info:24.3.5
podman push occne-repo-host:5000/occnp/oc-perf-info:24.3.5

podman tag docker.io/occnp/oc-diam-gateway:24.3.5 occne-repo-host:5000/
occnp/oc-diam-gateway:24.3.5
podman push occne-repo-host:5000/occnp/oc-diam-gateway:24.3.5

podman tag docker.io/occnp/oc-diam-connector:24.3.5 occne-repo-host:5000/
occnp/oc-diam-connector:24.3.5
podman push occne-repo-host:5000/occnp/oc-diam-connector:24.3.5

podman tag docker.io/occnp/oc-pcf-user:24.3.0 occne-repo-host:5000/occnp/oc-
pcf-user:24.3.0
podman push occne-repo-host:5000/occnp/oc-pcf-user:24.3.0

podman tag docker.io/occnp/ocdebug-tools:24.3.1 occne-repo-host:5000/occnp/
ocdebug-tools:24.3.1
podman push occne-repo-host:5000/occnp/ocdebug-tools:24.3.1

podman tag docker.io/occnp/oc-config-mgmt:24.3.5 occne-repo-host:5000/
occnp/oc-config-mgmt:24.3.5
podman push occne-repo-host:5000/occnp/oc-config-mgmt:24.3.5

podman tag docker.io/occnp/oc-config-server:24.3.5 occne-repo-host:5000/
occnp/oc-config-server:24.3.5
podman push occne-repo-host:5000/occnp/oc-config-server:24.3.5

podman tag docker.io/occnp/ocegress_gateway:24.3.3  occne-repo-host:5000/
occnp/ocegress_gateway:24.3.3
podman push occne-repo-host:5000/occnp/ocegress_gateway:24.3.3

podman tag docker.io/occnp/nrf-client:24.3.2 occne-repo-host:5000/occnp/nrf-
client:24.3.2
podman push occne-repo-host:5000/occnp/nrf-client:24.3.2

podman tag docker.io/occnp/common_config_hook:24.3.3 occne-repo-host:5000/
occnp/common_config_hook:24.3.3
podman push occne-repo-host:5000/occnp/common_config_hook:24.3.3

podman tag docker.io/occnp/configurationinit:24.3.3 occne-repo-host:5000/
occnp/configurationinit:24.3.3
podman push occne-repo-host:5000/occnp/configurationinit:24.3.3

podman tag docker.io/occnp/configurationupdate:24.3.3 occne-repo-host:5000/
occnp/configurationupdate:24.3.3
podman push occne-repo-host:5000/occnp/configurationupdate:24.3.3

podman tag docker.io/occnp/oc-soap-connector:24.3.0 occne-repo-host:5000/
occnp/occnp/oc-soap-connector:24.3.0
podman push occne-repo-host:5000/occnp/occnp/oc-soap-connector:24.3.0

podman tag docker.io/occnp/oc-pcrf-core:24.3.0 occne-repo-host:5000/occnp/
occnp/oc-pcrf-core:24.3.0
podman push occne-repo-host:5000/occnp/occnp/oc-pcrf-core:24.3.0
```

```
podman tag docker.io/occnp/oc-binding:24.3.0 occne-repo-host:5000/occnp/
occnp/oc-binding:24.3.0
podman push occne-repo-host:5000/occnp/occnp/oc-binding:24.3.0

podman tag docker.io/occnp/oc-bulwark:24.3.0 occne-repo-host:5000/occnp/
occnp/oc-bulwark:24.3.0
podman push occne-repo-host:5000/occnp/occnp/oc-bulwark:24.3.0

podman tag docker.io/occnp/oc-notifier:24.3.0 occne-repo-host:5000/occnp/
occnp/oc-notifier:24.3.0
podman push occne-repo-host:5000/occnp/occnp/oc-notifier:24.3.0

podman tag docker.io/occnp/oc-usage-mon:24.3.0 occne-repo-host:5000/occnp/
occnp/oc-usage-mon:24.3.0
podman push occne-repo-host:5000/occnp/occnp/oc-usage-mon:24.3.0

podman tag docker.io/occnp/oc-nwdaf-agent:24.3.0 occne-repo-host:5000/occnp/
occnp/
oc-nwdaf-agent:24.3.0
podman push occne-repo-host:5000/occnp/occnp/
oc-nwdaf-agent:24.3.0
```

2.2.1.3 Verifying and Creating Namespace

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

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

To verify and create a namespace:

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 "Creating Service Account, Role and RoleBinding.
2. If the required namespace is not available, create a namespace using the following command:

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

   Where,

   <required namespace> is the name of the namespace.

For example, the following command creates the namespace, `occnp`:

```
kubectl create namespace occnp
```

Sample output:

```
namespace/occnp created
```

3. Update the `global.nameSpace` parameter in `occnp_custom_values_24.3.0.yaml` file with the namespace created in step 2:
   Here is a sample configuration snippet from the `occnp_custom_values_24.3.0.yaml` file:

```
global:
  #NameSpace where secret is deployed
  nameSpace: occnp
```

Naming Convention for Namespace

The namespace should meet the following requirements:

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

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

2.2.1.4 Creating Service Account, Role and RoleBinding

This section is optional and it describes how to manually create a service account, role, and rolebinding resources. It is required only when customer needs to create a role, rolebinding, and service account manually before installing Policy.

> **Note**
>
> The secret(s) should exist in the same namespace where Policy is getting deployed. This helps to bind the Kubernetes role with the given service account.

Create Service Account, Role and RoleBinding

1. Run the following command to define the global service account by creating a Policy service account resource file:

```
vi <occnp-resource-file>
```

Example:

```
vi occnp-resource-template.yaml
```

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

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

A sample template to update the `occnp-resource-template.yaml` file is given below:

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

#---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: <helm-release>-role
  namespace: <namespace>
rules:
   - apiGroups:
       - ""
     resources:
       - services
       - configmaps
       - pods
       - secrets
       - endpoints
       - nodes
       - events
       - persistentvolumeclaims
     verbs:
       - get
       - list
       - watch
   - apiGroups:
       - apps
     resources:
       - deployments
       - statefulsets
     verbs:
       - get
       - watch
       - list
   - apiGroups:
       - autoscaling
```

```
      resources:
        - horizontalpodautoscalers
      verbs:
        - get
        - watch
        - list
#----

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

Where,

`<helm-release>` is a name provided by the user to identify the Policy Helm deployment.

`<namespace>` is a name provided by the user to identify the Kubernetes namespace of Policy. All the Policy microservices are deployed in this Kubernetes namespace.

> **Note**
>
> If you are installing Policy 22.1.0 using CNE 22.2.0 or later versions change the apiVersion of `kind:rolebinding` from `rbac.authorization.k8s.io/v1beta1` to `rbac.authorization.k8s.io/v1`.

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

   ```
   kubectl -n <namespace> create -f <occnp-resource-file>
   ```

   Example:

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

4. Update the `serviceAccountName` parameter in the `occnp_custom_values_24.3.0.yaml` file with the value updated in `name` field under `kind:ServiceAccount`. For more information about `serviceAccountName` parameter, see the "Configuration for Mandatory Parameters".

> **Note**
>
> `PodSecurityPolicy` kind is required for Pod Security Policy service account. For more information, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.

2.2.1.5 Creating Service Account, Role and Role Binding for Helm Test

This section describes the procedure to create service account, role, and role binding resources for Helm Test.

> **Important**
>
> The steps described in this section are optional and users may skip it in any of the following scenarios:
>
> - If user wants service accounts to be created automatically at the time of deploying CNC Policy
> - Global service account with associated role and role-bindings is already configured or the user has any in-house procedure to create service accounts.

Create Service Account

To create the global service account, create a YAML file (`occnp-sample-helmtestserviceaccount-template.yaml`) using the following sample code:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: <helm-release>-helmtestserviceaccount
  namespace: <namespace>
```

where `<helm-release>` is a name provided by the user to identify the helm deployment.

`namespace` is a name provided by the user to identify the Kubernetes namespace of the Policy. All the Policy microservices are deployed in this Kubernetes namespace.

Define Role Permissions

To define permissions using roles for Policy namespace, create a YAML file (`occnp-sample-role-template.yaml`) using the following sample code:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: <helm-release>-role
  namespace: <namespace>
rules:
   - apiGroups:
       - ""
     resources:
        -pods
```

```
-persistentvolumeclaims
-services
-endpoints
-configmaps
-events
-secrets
-serviceaccounts
verbs:
-list
-get
-watch
-apiGroups:
-apps
resources:
-deployments
-statefulsets
verbs:
-get
-watch
-list
-apiGroups:
-autoscaling
resources:
-horizontalpodautoscalers
verbs:
-get
-watch
-list
-apiGroups:
-policy
resources:
-poddisruptionbudgets
verbs:
-get
-watch
-list
-apiGroups:
-rbac.authorization.k8s.io
resources:
-roles
-rolebindings
verbs:
-get
-watch
-list
```

Creating Role Binding Template

To bind the above role with the service account, you must create role binding. To do so, create a YAML file (`occnp-sample-rolebinding-template.yaml`) using the following sample code:

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: RoleBinding
metadata:
  name: <helm-release>-rolebinding
```

```
  namespace: <namespace>
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: <helm-release>-role
subjects:
- kind: ServiceAccount
  name:  <helm-release>-helmtestserviceaccount
  namespace: <namespace>
```

Create resources

Run the following commands to create resources:

```
kubectl -n <namespace> create -f occnp-sample-helmtestserviceaccount-
template.yaml;
kubectl -n <namespace> create -f occnp-sample-role-template.yaml;
kubectl -n <namespace> create -f occnp-sample-rolebinding-template.yaml
```

> **Note**
>
> Once the global service account is added, users must add `global.helmTestServiceAccountName` in the `custom-values.yaml` file. Otherwise, installation can fail as a result of creating and deleting Custom Resource Definition (CRD).

2.2.1.6 Configuring Database, Creating Users, and Granting Permissions

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

Policy has four databases (Provisional, State, Release, Leaderpod, and NRF Client Database) and two users (Application and Privileged).

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

Policy Databases

For Policy applications, four types of databases are required:

1. **Provisional Database**: Provisional Database contains configuration information. The same configuration must be done on each site by the operator. Both Privileged User and Application User have access to this database. In case of georedundant deployments, each site must have a unique Provisional Database. Policy sites can access only the information in their unique Provisional Database.
   For example:

- For Site 1: occnp_config_server_site1
- For Site 2: occnp_config_server_site2
- For Site 3: occnp_config_server_site3

2. **State Database**: This database maintains the running state of Policy sites and has information of subscriptions, pending notification triggers, and availability data. It is replicated and the same configuration is maintained by all Policy georedundant sites. Both Privileged User and Application User have access to this database.
3. **Release Database**: This database maintains release version state, and it is used during upgrade and rollback scenarios. Only Privileged User has access to this database.
4. **Leaderpod Database**: This database is used to store leader and follower if PDB is enabled for microservices that require a single pod to be up in all the instances. The configuration of this database must be done on each site. In case of georedundant deployments, each site must have a unique Leaderpod database.
   For example:
   - For Site 1: occnp_leaderPodDb_site1
   - For Site 2: occnp_leaderPodDb_site2
   - For Site 3: occnp_leaderPodDb_site3

   > **Note**
   >
   > This database is used only when `nrf-client-nfmanagement.enablePDBSupport` is set to `true` in the `occnp_custom_values_24.3.0.yaml`. For more information, see NRF Client Configuration
5. **NRF Client Database**: This database is used to store discovery cache tables, and it also supports NRF Client features. Only Privileged User has access to this database and it is used only when the caching feature is enabled. In case of georedundant deployments, each site must have a unique NRF Client database and its configuration must be done on each site.
   For example:
   - For Site 1: occnp_nrf_client_site1
   - For Site 2: occnp_nrf_client_site2
   - For Site 3: occnp_nrf_client_site3

Policy Users

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

1. **Privileged User** : This user has a complete set of permissions. This user can perform create, alter, or drop operations on tables to perform install, upgrade, rollback, or delete operations.

   > **Note**
   >
   > In examples given in this document, Privileged User's username is '`occnpadminusr`' and password is '`occnpadminpasswd`'.
2. **Application User**: This user has a limited set of permissions and is used by Policy application to handle service operations. This user can create, insert, update, get, or remove the records. This user will not be able to alter, or drop the database or tables.

> **Note**
>
> In examples given in this document, Application User's username is 'occnpusr' and password is 'occnppasswd'.

This table lists the default database names and applicable deployment modes for various databases that need to be configured while deploying Policy:

Table 2-7 Policy Default Database Names

Service Name	Default Database Name	Database Type	Applicable for
SM Service	occnp_pcf_sm	State	Converged Policy and PCF
AM Service	occnp_pcf_am	State	Converged Policy and PCF
PCRF Core Service	occnp_pcrf_core	State	Converged Policy and PCRF
Binding Service	occnp_binding	State	Converged Policy and PCF
PDS Service	occnp_policyds	State	Converged Policy and PCF
UE Service	occnp_pcf_ue	State	Converged Policy and PCF
CM Service	occnp_commonconfig occnp_cmservice	Provisional	Converged Policy, PCF, and PCRF
Config Server Service	occnp_config_server	Provisional	Converged Policy, PCRF, and PCF
Audit Service	occnp_audit_service	Provisional	Converged Policy and PCF
Usage Monitoring	occnp_usagemon	Provisional	Converged Policy, PCF, and PCRF
NRF Client	occnp_nrf_client occnp_leaderPodDb	Provisional	Converged Policy and PCF
NWDAF Agent	occnp_pcf_nwdaf_agent	Provisional	Converged Policy and PCF
Perf Info Service	occnp_overload	Leaderpod	Converged Policy, PCRF, and PCF
Release	occnp_release	Release	

2.2.1.6.1 Single Site

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

1. Log in to the machine where SSH keys are stored and have permission to access the SQL nodes of NDB cluster.
2. Connect to the SQL nodes.
3. Log in to the MySQL prompt using root permission, or log in as a user who has the permission to create users as per conditions explained in the next step.

Example:

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

> **Note**
>
> This command varies between systems, path for MySQL binary, root user, and root password. After running this command, enter the password specific to the user mentioned in the command.

4. Run the following command to check if both the Policy users already exist:

   ```
   SELECT User FROM mysql.user;
   ```

   If the users already exist, go to the next step. Else, create the respective new user or users by following the steps below:

   - Run the following command to create a new Privileged User:

     ```
     CREATE USER '<Policy Privileged-User Name>'@'%' IDENTIFIED BY '<Policy
     Privileged-User Password>';
     ```

     Example:

     ```
     CREATE USER 'occnpadminusr'@'%' IDENTIFIED BY 'occnpadminpasswd';
     ```

   - Run the following command to create a new Application User:

     ```
     CREATE USER '<Application User Name>'@'%' IDENTIFIED BY '<APPLICATION
     Password>';
     ```

     Example:

     ```
     CREATE USER 'occnpusr'@'%' IDENTIFIED BY 'occnppasswd';
     ```

5. Run the following command to check whether any of the Policy database already exists:

   ```
   show databases;
   ```

   a. If any of the previously configured database is already present, remove them. Otherwise, skip this step.
   Run the following command to remove a preconfigured Policy database:

      ```
      DROP DATABASE if exists <DB Name>;
      ```

      Example:

      ```
      DROP DATABASE if exists occnp_audit_service;
      ```

b. Run the following command to create a new Policy database if it does not exist, or after dropping an existing database:

```
CREATE DATABASE IF NOT EXISTS <DB Name> CHARACTER SET utf8;
```

For example: Sample illustration for creating all database required for Policy installation.

```
CREATE DATABASE IF NOT EXISTS occnp_policyds CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_audit_service CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_config_server CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_am CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_sm CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcrf_core CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_release CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_binding CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_ue CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_commonconfig CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_cmservice CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_usagemon CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_overload CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_nwdaf_agent CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_leaderPodDb CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_nrf_client CHARACTER SET utf8;
```

> **Note**
>
> Ensure that you use the same database names while creating database that you have used in the global parameters of `occnp_custom_values_24.3.0.yaml` file. Following is an example of what are the names of the policy database names configured in the `occnp_custom_values_24.3.0.yaml` file:

```
global:
  releaseDbName: occnp_release
  nrfClientDbName: occnp_nrf_client
policyds:
  envMysqlDatabase: occnp_policyds
audit-service:
  envMysqlDatabase: occnp_audit_service
config-server:
  envMysqlDatabase:occnp_config_server
am-service:
  envMysqlDatabase: occnp_pcf_am
sm-service:
  envMysqlDatabase: occnp_pcf_sm
pcrf-core:
  envMysqlDatabase: occnp_pcrf_core
binding:
  envMysqlDatabase: occnp_binding
ue-service:
  envMysqlDatabase: occnp_pcf_ue
cm-service:
  envMysqlDatabase: occnp_cmservice
usage-mon:
  envMysqlDatabase: occnp_usagemon
nwdaf-agent:
  envMysqlDatabase: occnp_pcf_nwdaf_agent
nrf-client-nfmanagement:
  dbConfig:
    leaderPodDbName: occnp_leaderPodDb
```

6. Grant permissions to users on the database:

> **Note**
>
> Creation of database is optional if grant is scoped to all database, that is, database name is not mentioned in grant command.

a. Run the following command to grant NDB_STORED_USER permissions to the Privileged User:

```
GRANT NDB_STORED_USER ON *.* TO 'occnpadminusr'@'%';
```

b. Run the following commands to grant Privileged User permission on all Policy Databases:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, ALTER,
CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE ON <DB Name>.*TO `<Policy
Privileged-User Name>`@`%`;
```

For example:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_pcf_sm.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_pcf_am.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_config_server.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_audit_service.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_release.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_pcrf_core.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_binding.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_policyds.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_pcf_ue.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_cmservice.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_commonconfig.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_overload.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_nrf_client.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_pcf_nwdaf_agent.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_usagemon.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP ON
mysql.ndb_replication TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, CREATE
TEMPORARY TABLES, DELETE, UPDATE, EXECUTE ON occnp_leaderPodDb.* TO
'occnpadminusr'@'%';
```

c. Run the following command to grant NDB_STORED_USER permissions to the Application User:

```
GRANT NDB_STORED_USER ON *.* TO 'occnpusr'@'%';
```

d. Run the following commands to grant Application User permission on all Policy Databases:

```
GRANT SELECT, INSERT, LOCK TABLES, DELETE, UPDATE, REFERENCES, EXECUTE
ON <DB Name>.* TO '<Application User Name>'@'%';
```

For example:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_pcf_sm.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_pcf_am.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_config_server.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_audit_service.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_pcrf_core.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_binding.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_policyds.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_pcf_ue.* TO
'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON occnp_commonconfig.* TO
'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON occnp_cmservice.* TO
'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON occnp_usagemon.* TO
'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON occnp_pcf_nwdaf_agent.*
TO 'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_overload.* TO
'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON occnp_nrf_client.* TO
'occnpusr'@'%';
```

7. Run the following command to verify that the privileged or application users have all the required permissions:

```
show grants for username;
```

where `username` is the name of the privileged or application user.

Example:

```
show grants for occnpadminusr;
show grants for occnpusr;
```

8. Run the following command to flush privileges:

```
FLUSH PRIVILEGES;
```

9. Exit from MySQL prompt and SQL nodes.

2.2.1.6.2 Multisite

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

For Policy georedundant deployment, listed databases names must be unique for each site. For the remaining databases, the database name must be same across all the sites.

It is recommended to use unique database names when multiple instances of Policy use and share a single cnDBtier (MySQL cluster) in the network. To maintain unique database names for all the NF instances in the network, a good practice is to add the deployment name of the Policy instance as a prefix or suffix to the database name. However, you can use any prefix or suffix to create the unique database name.

> **Note**
>
> Before running the procedure for georedundant sites, ensure that the cnDBTier for georedundant sites is up and replication channels are enabled.

Table 2-8 Policy Unique Databases names for two site and three site deployment

Two Site Database Names	Three Site Database Names
occnp_config_server_site1 occnp_config_server_site2	occnp_config_server_site1 occnp_config_server_site2 occnp_config_server_site3
occnp_cmservice_site1 occnp_cmservice_site2	occnp_cmservice_site1 occnp_cmservice_site2 occnp_cmservice_site3
occnp_commonconfig_site1 occnp_commonconfig_site2	occnp_commonconfig_site1 occnp_commonconfig_site2 occnp_commonconfig_site3
occnp_leaderPodDb_site1 occnp_leaderPodDb_site2	occnp_leaderPodDb_site1 occnp_leaderPodDb_site2 occnp_leaderPodDb_site3
occnp_overload_site1 occnp_overload_site2	occnp_overload_site1 occnp_overload_site2 occnp_overload_site3
occnp_audit_service_site1 occnp_audit_service_site2	occnp_audit_service_site1 occnp_audit_service_site2 occnp_audit_service_site3
occnp_pcf_nwdaf_agent_site1 occnp_pcf_nwdaf_agent_site2	occnp_pcf_nwdaf_agent_site1 occnp_pcf_nwdaf_agent_site2 occnp_pcf_nwdaf_agent_site3
occnp_nrf_client_site1 occnp_nrf_client_site2	occnp_nrf_client_site1 occnp_nrf_client_site2 occnp_nrf_client_site3

1. Log in to the machine where SSH keys are stored and have permission to access the SQL nodes of NDB cluster.
2. Connect to the SQL nodes.
3. Log in to the MySQL prompt using root permission, or log in as a user who has the permission to create users as per conditions explained in the next step.
   Example:

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

   > **Note**
   >
   > This command varies between systems, path for MySQL binary, root user, and root password. After running this command, enter the password specific to the user mentioned in the command.

4. Run the following command to check if both the Policy users already exist:

   ```
   SELECT User FROM mysql.user;
   ```

   If the users already exist, go to the next step. Otherwise, create the respective new user or users by following the steps below:

   - Run the following command to create a new Privileged User:

     ```
     CREATE USER '<Policy Privileged-User Name>'@'%' IDENTIFIED BY '<Policy
     Privileged-User Password>';
     ```

     Example:

     ```
     CREATE USER 'occnpadminusr'@'%' IDENTIFIED BY 'occnpadminpasswd';
     ```

   - Run the following command to create a new Application User:

     ```
     CREATE USER '<Application User Name>'@'%' IDENTIFIED BY '<APPLICATION
     Password>';
     ```

     Example:

     ```
     CREATE USER 'occnpusr'@'%' IDENTIFIED BY 'occnppasswd';
     ```

   > **Note**
   >
   > You must create both the users on all the SQL nodes for all georedundant sites.

5. Run the following command to check whether any of the Policy database already exists:

   ```
   show databases;
   ```

a. If any of the previously configured database is already present, remove them. Otherwise, skip this step.

> ⚠ **Caution**
>
> In case you have georedundant sites configured, removal of the database from any one of the SQL nodes of any cluster will remove the database from all georedundant sites.

Run the following command to remove a preconfigured Policy database:

```
DROP DATABASE if exists <DB Name>;
```

Example:

```
DROP DATABASE if exists occnp_audit_service;
```

b. Run the following command to create a new Policy database if it does not exist, or after dropping an existing database:

```
CREATE DATABASE IF NOT EXISTS <DB Name> CHARACTER SET utf8;
```

For example: Sample illustration for creating all database required for Policy installation in site1.

```
CREATE DATABASE IF NOT EXISTS occnp_config_server_site1 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_cmservice_site1 CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_commonconfig_site1 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_leaderPodDb_site1 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_overload_site1 CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_audit_service_site1 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_nwdaf_agent_site1 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_policyds CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_am CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_sm CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcrf_core CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_release CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_binding CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_ue CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_usagemon CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_nrf_client_site1 CHARACTER SET utf8;
```

For example: Sample illustration for creating all database required for Policy installation in site2.

```
CREATE DATABASE IF NOT EXISTS occnp_config_server_site2 CHARACTER SET
utf8;
```

```
CREATE DATABASE IF NOT EXISTS occnp_cmservice_site2 CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_commonconfig_site2 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_leaderPodDb_site2 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_overload_site2 CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_audit_service_site2 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_nwdaf_agent_site2 CHARACTER SET
utf8;
CREATE DATABASE IF NOT EXISTS occnp_policyds CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_am CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_sm CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcrf_core CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_release CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_binding CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_pcf_ue CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_usagemon CHARACTER SET utf8;
CREATE DATABASE IF NOT EXISTS occnp_nrf_client_site2 CHARACTER SET utf8;
```

> **Note**
>
> Ensure that you use the same database names while creating database that you have used in the global parameters of `occnp_custom_values_24.3.0.yaml` files. Following is an example of what are the names of the Policy database names configured in the `occnp_custom_values_24.3.0.yaml` files in site1 and site2:

```
global:
  nrfClientDbName: occnp_nrf_client_site1
audit-service:
  envMysqlDatabase: occnp_audit_service_site1
config-server:
  envMysqlDatabase:occnp_config_server_site1
cm-service:
  envMysqlDatabase: occnp_cmservice_site1
nwdaf-agent:
  envMysqlDatabase: occnp_pcf_nwdaf_agent_site1
nrf-client-nfmanagement:
  dbConfig:
    leaderPodDbName: occnp_leaderPodDb_site1
```

```
global:
  nrfClientDbName: occnp_nrf_client_site2
audit-service:
  envMysqlDatabase: occnp_audit_service_site2
config-server:
  envMysqlDatabase:occnp_config_server_site2
cm-service:
  envMysqlDatabase: occnp_cmservice_site2
nwdaf-agent:
  envMysqlDatabase: occnp_pcf_nwdaf_agent_site2
nrf-client-nfmanagement:
  dbConfig:
    leaderPodDbName: occnp_leaderPodDb_site2
```

6. Grant permissions to users on the database:

> **Note**
>
> - Run this step on all the SQL nodes for each Policy standalone site in a georedundant deployment.
> - Creation of database is optional if grant is scoped to all database, that is, database name is not mentioned in grant command.

a. Run the following command to grant NDB_STORED_USER permissions to the Privileged User:

```
GRANT NDB_STORED_USER ON *.* TO 'occnpadminusr'@'%';
```

b. Run the following commands to grant Privileged User permission on all Policy Databases:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, ALTER,
CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE ON <DB Name>.*TO `<Policy
Privileged-User Name>`@`%`;
```

For example for site1:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_config_server_site1.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_audit_service_site1.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_cmservice_site1.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_commonconfig_site1.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_overload_site1.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_pcf_nwdaf_agent_site1.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_nrf_client_site1.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, CREATE
TEMPORARY TABLES, DELETE, UPDATE, EXECUTE ON occnp_leaderPodDb_site1.*
TO 'occnpadminusr'@'%';
```

For example for site2:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_config_server_site2.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_audit_service_site2.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_cmservice_site2.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_commonconfig_site2.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_overload_site2.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_pcf_nwdaf_agent_site2.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, REFERENCES,
INDEX ON occnp_nrf_client_site2.* TO 'occnpadminusr'@'%';
GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, CREATE
TEMPORARY TABLES, DELETE, UPDATE, EXECUTE ON occnp_leaderPodDb_site2.*
TO 'occnpadminusr'@'%';
```

c. Run the following command to grant NDB_STORED_USER permissions to the Application User:

```
GRANT NDB_STORED_USER ON *.* TO 'occnpusr'@'%';
```

d. Run the following commands to grant Application User permission on all Policy Databases:

```
GRANT SELECT, INSERT, LOCK TABLES, DELETE, UPDATE, REFERENCES, EXECUTE
ON <DB Name>.* TO '<Application User Name>'@'%';
```

For example in Policy site1:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_cmservice_site1.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_config_server_site1.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_audit_service_site1.* TO
'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON
occnp_commonconfig_site1.* TO 'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON
occnp_pcf_nwdaf_agent_site1.* TO 'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_overload_site1.*
TO 'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON
occnp_nrf_client_site1.* TO 'occnpusr'@'%';
```

For example in Policy site2:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_cmservice_site2.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_config_server_site2.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_audit_service_site2.* TO
'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON
occnp_commonconfig_site2.* TO 'occnpusr'@'%';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON
occnp_pcf_nwdaf_agent_site2.* TO 'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_overload_site2.*
TO 'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON
occnp_nrf_client_site2.* TO 'occnpusr'@'%';
```

7. Run the following command to verify that the privileged or application users have all the required permissions:

```
show grants for username;
```

where `username` is the name of the privileged or application user.

Example:

```
show grants for occnpadminusr;
show grants for occnpusr;
```

8. Run the following command to flush privileges:

   ```
   FLUSH PRIVILEGES;
   ```

9. Exit from MySQL prompt and SQL nodes.

2.2.1.7 Configuring Kubernetes Secret for Accessing Database

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

2.2.1.7.1 Creating and Updating Secret for 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=mysql-username=<Privileged Mysql database username> --from-
   literal=mysql-password=<Privileged Mysql User database passsword> -n
   <Namespace>
   ```

   Where,

   `<Privileged User secret name>` is the secret name of the Privileged User.

   `<Privileged MySQL database username>` is the username of the Privileged User.

   `<Privileged MySQL User database passsword>` is the password of the Privileged User.

   `<Namespace>` is the namespace of Policy 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 occnp-privileged-db-pass --from-
   literal=mysql-username=occnpadminusr --from-literal=mysql-
   password=occnpadminpasswd -n occnp
   ```

2. Run the following command to verify the secret created:

   ```
   kubectl describe secret <Privileged User secret name> -n <Namespace>
   ```

   Where,

<Privileged User secret name> is the secret name of the database.

<Namespace> is the namespace of Policy deployment.

For example:

```
kubectl describe secret occnp-privileged-db-pass -n occnp
```

Sample output:

```
Name:  occnp-privileged-db-pass
Namespace:    occnp
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
mysql-password:  10 bytes
mysql-username:  17 bytes
```

3. Update the command used in step 1 with string "`--dry-run -o yaml`" and "

```
kubectl replace -f - -n <Namespace of Policy deployment>
```

". After the update is performed, use the following command:

```
kubectl create secret generic <Privileged User secret name> --from-
literal=mysql-username=<Privileged MySQL database username> --from-
literal=mysql-password=<Privileged Mysql database password> --dry-run -o
yaml -n <Namespace> | kubectl replace -f - -n <Namespace>
```

Where,

<Privileged User secret name> is the secret name of the Privileged User.

<Privileged MySQL database username> is the username of the Privileged User.

<Privileged MySQL User database passsword> is the password of the Privileged User.

<Namespace> is the namespace of Policy deployment.

4. Run the updated command. The following message is displayed:
`secret/<Privileged User secret name> replaced`

Where,

<Privileged User secret name> is the updated secret name of the Privileged User.

2.2.1.7.2 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.

1. Run the following command to create Kubernetes secret:

   ```
   kubectl create secret generic <Application User secret name> --from-
   literal=mysql-username=<Application MySQL Database Username> --from-
   literal=mysql-password=<Application MySQL User database passsword> -n
   <Namespace>
   ```

   Where,

   `<Application User secret name>` is the secret name of the Application User.

   `<Application MySQL database username>` is the username of the Application User.

   `<Application MySQL User database passsword>` is the password of the Application User.

   `<Namespace>` is the namespace of Policy 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 occnp-db-pass --from-literal=mysql-
   username=occnpusr --from-literal=mysql-password=occnppasswd -n occnp
   ```

2. Run the following command to verify the secret created:

   ```
   kubectl describe secret <Application User secret name> -n <Namespace>
   ```

   Where,

   `<Application User secret name>` is the secret name of the database.

   `<Namespace>` is the namespace of Policy deployment.

   For example:

   ```
   kubectl describe secret occnp-db-pass -n occnp
   ```

   Sample output:

   ```
   Name:  occnp-db-pass
   Namespace:    occnp
   Labels:       <none>
   Annotations:  <none>

   Type:  Opaque

   Data
   ====
   mysql-password:  10 bytes
   mysql-username:  17 bytes
   ```

3. Update the command used in step 1 with string "`--dry-run -o yaml`" and "

   ```
   kubectl replace -f - -n <Namespace of Policy deployment>
   ```

   ". After the update is performed, use the following command:

   ```
   kubectl create secret generic <Application User secret name> --from-
   literal=mysql-username=<Application MySQL database username> --from-
   literal=mysql-password=<Application Mysql database password> --dry-run -o
   yaml -n <Namespace> | kubectl replace -f - -n <Namespace>
   ```

   Where,

   `<Application User secret name>` is the secret name of the Application User.

   `<Application MySQL database username>` is the username of the Application User.

   `<Application MySQL User database passsword>` is the password of the Application User.

   `<Namespace>` is the namespace of Policy deployment.

4. Run the updated command. The following message is displayed:
   `secret/<Application User secret name> replaced`

   Where,

   `<Application User secret name>` is the updated secret name of the Application User.

2.2.1.7.3 Creating Secret for Support of TLS in Diameter Gateway

This section explains how to create Kubernetes secret to store private key, public key, and trust chain certificates to support TLS in Diameter Gateway.

1. Run the following command to create Kubernetes secret:

   ```
   kubectl create secret generic <TLS_SECRET_NAME> --from-
   file=<TLS_RSA_PRIVATE_KEY_FILENAME/TLS_ECDSA_PRIVATE_KEY_FILENAME> --from-
   file=<TLS_CA_BUNDLE_FILENAME> --from-file=<TLS_RSA_CERTIFICATE_FILENAME/
   TLS_ECDSA_CERTIFICATE_FILENAME> -n <Namespace of OCCNP deployment>.
   ```

   For example:

   ```
   kubectl create secret generic dgw-tls-secret --from-file=dgw-key.pem --
   from-file=ca-cert.cer --from-file=dgw-cert.crt -n vega-ns6
   ```

   Where,

   `dgw-key.pem` is the private Key of diam-gateway (either generated by RSA or ECDSA).

   `dgw-cert.crt` is the public Key certificate of diam-gateway (either generated by RSA or ECDSA).

   `ca-cert.cer` is the trust Chain Certificate file, either an Intermediate CA or Root CA.

   `dgw-tls-secret` is the default name of the secret.

2.2.1.8 Enabling MySQL based DB Compression

If you are using Policy 22.4.5 or later versions, you must enable MySQL based DB Compression by adding the following configurations in the custom-values.yaml file at the time of installation or upgrade:

```
mySqlDbCompressionEnabled: 'true'
mySqlDbCompressionScheme: '1'
```

> **Note**
>
> Data compression must be activated when all sites are upgraded to 22.4.5. Rollback is not possible once data compression is activated.

For more information on DB compression configurations, see PCRF-Core.

2.2.1.9 Configuring Secrets for Enabling HTTPS

This section explains the steps to create and update the Kubernetes secret and enable HTTPS at Egress Gateway.

2.2.1.9.1 Managing HTTPS at Ingress Gateway

This section explains the steps to configure secrets for enabling HTTPS in Ingress Gateway. This procedure must be performed before deploying Policy.

Creating and Updating Secrets at Ingress Gateway

> **Note**
>
> The passwords for TrustStore and KeyStore are stored in respective password files. The process to create private keys, certificates, and passwords is at the discretion of the user or operator. To create Kubernetes secret for HTTPS, following files are required:
>
> - PCF Private Key and Certificate (either generated by RSA or ECDSA)
> - Trust Chain Certificate file, either an Intermediate CA or Root CA
> - TrustStore password file

1. Run the following command to create secret:

   ```
   kubectl create secret generic <ocingress-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> -n <Namespace of OCCNP deployment>
   ```

   Where,

<ocingress-secret-name> is the secret name for Ingress Gateway.

<ssl_ecdsa_private_key.pem> is the ECDSA private key.

<rsa_private_key_pkcs1.pem> is the RSA private key.

<ssl_truststore.txt> is the SSL Truststore file.

<caroot.cer> is the CA root file.

<ssl_rsa_certificate.crt> is the SSL RSA certificate.

<ssl_ecdsa_certificate.crt> is the SSL ECDSA certificate.

<Namespace> of Policy deployment.

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

For example:

```
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 -n occnp
```

> **Note**
>
> It is recommended to use the same secret name as mentioned in the example. In case you change <ocingress-secret-name>, update the k8SecretName parameter under ingressgateway attributes section in the occnp_custom_values_24.3.0.yaml file.

2. Run the following command to verify the details of the secret created:

```
kubectl describe secret <ocingress-secret-name> -n <Namespace of OCCNP
deployment>
```

Where,

<ocingress-secret-name> is the secret name for Ingress Gateway.

Namespace of Policy deployment.

For example:

```
kubectl describe secret ocingress-secret -n occnp
```

3. Update the command used in Step 1 with string "`--dry-run -o yaml`" and "`kubectl replace -f - -n <Namespace of occnp deployment>`". After the update is performed, use the following command:

   ```
   kubectl create secret generic <ocingress-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> |
   kubectl replace -f - -n <Namespace>
   ```

   For example:

   ```
   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 occnp | kubectl
   replace -f - -n occnp
   ```

   > **Note**
   >
   > The names used in the aforementioned command must be same as the names provided in the `occnp_custom_values_24.3.0.yaml` in Policy deployment.

4. Run the updated command.
5. After the secret update is complete, the following message appears:
   `secret/<ocingress-secret> replaced`

Enabling HTTPS at Ingress Gateway

This step is required only when SSL settings needs to be enabled on Ingress Gateway microservice of Policy.

1. Enable `enableIncomingHttps` parameter under Ingress Gateway Global Parameters section in the `occnp-24.3.0-custom-values-occnp.yaml` file. For more information about `enableIncomingHttps` parameter, see under global parameters section of the `occnp-24.3.0-custom-values-occnp.yaml` file.
2. Configure the following details in the `ssl` section under `ingressgateway attributes`, in case you have changed the attributes while creating secret:
   - Kubernetes namespace
   - Kubernetes secret name holding the certificate details
   - Certificate information

   ```
   ingress-gateway:
     # ---- HTTPS Configuration - BEGIN ----
     enableIncomingHttps: true

     service:
       ssl:
         privateKey:
   ```

```
    k8SecretName: occnp-gateway-secret
    k8NameSpace: occnp
    rsa:
      fileName: rsa_private_key_pkcs1.pem
  certificate:
    k8SecretName: occnp-gateway-secret
    k8NameSpace: occnp
    rsa:
      fileName: ocegress.cer
  caBundle:
    k8SecretName: occnp-gateway-secret
    k8NameSpace: occnp
    fileName: caroot.cer
  keyStorePassword:
    k8SecretName: occnp-gateway-secret
    k8NameSpace: occnp
    fileName: key.txt
  trustStorePassword:
    k8SecretName: occnp-gateway-secret
    k8NameSpace: occnp
    fileName: trust.txt
```

3. Save the `occnp_custom_values_24.3.0.yaml` file.

2.2.1.9.2 Managing HTTPS at Egress Gateway

This section explains the steps to create and update the Kubernetes secret and enable HTTPS at Egress Gateway.

Creating and Updating Secrets at Egress Gateway

> **Note**
>
> The passwords for TrustStore and KeyStore are stored in respective password files. The process to create private keys, certificates, and passwords is at the discretion of the user or operator. To create Kubernetes secret for HTTPS, following files are required:
>
> - PCF Private Key and Certificate (either generated by RSA or ECDSA)
> - Trust Chain Certificate file, either an Intermediate CA or Root CA
> - TrustStore password file

1. Run the following command to create secret:

   ```
   kubectl create secret generic <ocegress-secret-name> --from-
   file=<ssl_ecdsa_private_key.pem> --from-file=<ssl_rsa_private_key.pem> --
   from-file=<ssl_truststore.txt> --from-file=<ssl_keystore.txt> --from-
   file=<ssl_cabundle.crt> --from-file=<ssl_rsa_certificate.crt> --from-
   file=<ssl_ecdsa_certificate.crt> -n <Namespace of OCCNP deployment>
   ```

   Where,

   `<ocegress-secret-name>` is the secret name for Egress Gateway.

<ssl_ecdsa_private_key.pem> is the ECDSA private key.

<rsa_private_key_pkcs1.pem> is the RSA private key.

<ssl_truststore.txt> is the SSL Truststore file.

<caroot.cer> is the CA root file.

<ssl_rsa_certificate.crt> is the SSL RSA certificate.

<ssl_ecdsa_certificate.crt> is the SSL ECDSA certificate.

<Namespace> of Policy deployment.

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

For example:

```
kubectl create secret generic ocegress-secret --from-
file=ssl_ecdsa_private_key.pem --from-file=ssl_rsa_private_key.pem --from-
file=ssl_truststore.txt --from-file=ssl_keystore.txt --from-
file=ssl_cabundle.crt --from-file=ssl_rsa_certificate.crt --from-
file=ssl_ecdsa_certificate.crt -n occnp
```

> **Note**
>
> It is recommended to use the same secret name as mentioned in the example. In case you change <ocegress-secret-name>, update the k8SecretName parameter under egressgateway attributes section in the occnp_custom_values_24.3.0.yaml file.

2. Run the following command to verify the details of the secret created:

   ```
   kubectl describe secret <ocegress-secret-name> -n <Namespace of OCCNP
   deployment>
   ```

   Where,

   <ocegress-secret-name> is the secret name for Egress Gateway.

   Namespace of Policy deployment.

   For example:

   ```
   kubectl describe secret ocegress-secret -n occnp
   ```

3. Update the command used in Step 1 with string "--dry-run -o yaml" and "kubectl replace -f - -n <Namespace of occnp deployment>". After the update is performed, use the following command:

   ```
   kubectl create secret generic <ocegress-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> |
kubectl replace -f - -n <Namespace>
```

For example:

```
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 occnp | kubectl
replace -f - -n occnp
```

> **Note**
>
> The names used in the aforementioned command must be same as the names provided in the `occnp_custom_values_24.3.0.yaml` in Policy deployment.

4. Run the updated command.
5. After the secret update is complete, the following message appears:
   `secret/<ocegress-secret> replaced`

Enabling HTTPS at Egress Gateway

This step is required only when SSL settings needs to be enabled on Egress Gateway microservice of Policy.

1. Enable `enableOutgoingHttps` parameter under egressgateway attributes section in the `occnp-24.3.0-custom-values-occnp.yaml` file. For more information about `enableOutgoingHttps` parameter, see the Egress Gateway section.
2. Configure the following details in the `ssl` section under `egressgateway attributes`, in case you have changed the attributes while creating secret:
   - Kubernetes namespace
   - Kubernetes secret name holding the certificate details
   - Certificate information

```
egress-gateway:
  #Enabling it for egress https requests
  enableOutgoingHttps: true

  service:
    ssl:
      privateKey:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        rsa:
          fileName: rsa_private_key_pkcs1.pem
        ecdsa:
          fileName: ssl_ecdsa_private_key.pem
      certificate:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
```

```
    rsa:
      fileName: ocegress.cer
    ecdsa:
      fileName: ssl_ecdsa_certificate.crt
  caBundle:
    k8SecretName: ocpcf-gateway-secret
    k8NameSpace: ocpcf
    fileName: caroot.cer
  keyStorePassword:
    k8SecretName: ocpcf-gateway-secret
    k8NameSpace: ocpcf
    fileName: key.txt
  trustStorePassword:
    k8SecretName: ocpcf-gateway-secret
    k8NameSpace: ocpcf
    fileName: trust.txt
```

3. Save the `occnp_custom_values_24.3.0.yaml` file.

2.2.1.10 Configuring Secrets to Enable Access Token

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

Generating KeyPairs for NRF Instances

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

Using the Openssl tool, user can generate KeyPairs for each of the NRF instances. The commands to generate the KeyPairs are as follow:

> **Note**
>
> Here, it is assumed that there are only two NRF instances with the the following instance IDs:
>
> - **NRF Instance 1**: 664b344e-7429-4c8f-a5d2-e7dfaaaba407
> - **NRF Instance 2**: 601aed2c-e314-46a7-a3e6-f18ca02faacc

Example Command to generate KeyPair for NRF Instance 1

```
Generate a 2048-bit RSA private key

openssl genrsa -out private_key.pem 2048
Convert private Key to PKCS#8 format (so Java can read it)

openssl pkcs8 -topk8 -inform PEM -outform PEM -in private_key.pem -out
private_key_pkcs.der -nocrypt
Output public key portion in PEM format (so Java can read it)
```

```
openssl rsa -in private_key.pem -pubout -outform PEM -out public_key.pem
Create reqs.conf and place the required content for NRF certificate
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
prompt = no
[req_distinguished_name]
C = IN
ST = BLR
L = TempleTerrace
O = Personal
CN = nnrf-001.tmtrflaa.5gc.tmp.com
[v3_req]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, dataEncipherment
subjectAltName = DNS:nnrf-001.tmtrflaa.5gc.tmp.com
#subjectAltName = URI:UUID:6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c
#subjectAltName = otherName:UTF8:NRF

Output ECSDA private key portion in PEM format and corresponding NRF
certificate in {nrfInstanceId}_ES256.crt file

openssl req -x509 -new -out {nrfInstanceId}_ES256.crt -newkey ec:<(openssl
ecparam -name secp521r1) -nodes -sha256 -keyout ecdsa_private_key.key -config
reqs.conf

#Replace the place holder "{nrfInstanceId}" with NRF Instance 1's UUID while
running the command.
Example below
$ openssl req -x509 -new -out 664b344e-7429-4c8f-a5d2-e7dfaaaba407_ES256.crt -
newkey ec:<(openssl ecparam -name secp521r1) -nodes -sha256 -keyout
ecdsa_private_key.key -config reqs.conf
```

The output is a set of Private Key and NRF Certificate similar to the following:

NRF1 (Private key: ecdsa_private_key.key, NRF Public Certificate: 664b344e-7429-4c8f-a5d2-e7dfaaaba407_ES256.crt)

Example Command to generate KeyPair for NRF Instance 2

```
Generate a 2048-bit RSA private key

openssl genrsa -out private_key.pem 2048
Convert private Key to PKCS#8 format (so Java can read it)

openssl pkcs8 -topk8 -inform PEM -outform PEM -in private_key.pem -out
private_key_pkcs.der -nocrypt
Output public key portion in PEM format (so Java can read it)

openssl rsa -in private_key.pem -pubout -outform PEM -out public_key.pem
Create reqs.conf and place the required content for NRF certificate
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
prompt = no
[req_distinguished_name]
```

```
C = IN
ST = BLR
L = TempleTerrace
O = Personal
CN = nnrf-001.tmtrflaa.5gc.tmp.com
[v3_req]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, dataEncipherment
subjectAltName = DNS:nnrf-001.tmtrflaa.5gc.tmp.com
#subjectAltName = URI:UUID:6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c
#subjectAltName = otherName:UTF8:NRF

Output ECSDA private key portion in PEM format and corresponding NRF
certificate in {nrfInstanceId}_ES256.crt file

openssl req -x509 -new -out {nrfInstanceId}_ES256.crt -newkey ec:<(openssl
ecparam -name prime256v1) -nodes -sha256 -keyout ecdsa_private_key.key -
config reqs.conf

#Replace the place holder "{nrfInstanceId}" with NRF Instance 2's UUID while
running the command.
Example below
$ openssl req -x509 -new -out 601aed2c-e314-46a7-a3e6-f18ca02faacc_ES256.crt -
newkey ec:<(openssl ecparam -name prime256v1) -nodes -sha256 -keyout
ecdsa_private_key.key -config reqs.conf
```

The output is a set of Private Key and NRF Certificate similar to the following:

NRF2 (Private key: ecdsa_private_key.key, PublicCerificate: 601aed2c-e314-46a7-a3e6-f18ca02faacc_ES256.crt)

Enabling and Configuring Access Token

To enable access token validation, configure both Helm-based and REST-based configurations on Ingress Gateway.

Configuration using Helm:

For Helm-based configuration, perform the following steps:

1. Create a Namespace for Secrets. The namespace is used as an input to create Kubernetes secret for private keys and public certificates. Create a namespace using the following command:

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

   Where,

   <required namespace> is the name of the namespace.

   For example, the following command creates the namespace, `ocpcf`:

   ```
   kubectl create namespace ocpcf
   ```

2. Create Kubernetes Secret for NRF Public Key. To create a secret using the Public keys of the NRF instances, run the following command:

   ```
   kubectl create secret generic <secret-name> --from-file=<filename.crt> -n
   <Namespace>
   ```

   Where,

   <secret-name> is the secret name.

   <Namespace> is the PCF namespace.

   <filename.crt> is the public key certificate and we can have any number of certificates in the secret.

   For example:

   ```
   kubectl create secret generic nrfpublickeysecret --from-file=./
   664b344e-7429-4c8f-a5d2-e7dfaaaba407_ES256.crt --from-file=./601aed2c-
   e314-46a7-a3e6-f18ca02faacc_ES256.crt -n ocpcf
   ```

   > **Note**
   >
   > In the above command:
   >
   > - `nrfpublickeysecret` is the secret name
   > - `ocpcf` is the namespace
   > - `.crt` files is the public key certificates

   We can have any number of certificates in the secret.

3. Enable Access token using Helm Configuration by setting the Ingress Gateway parameter `oauthValidatorEnabled` parameter value to `true`.
   Further, configure the secret and namespace on Ingress Gateway in the OAUTH CONFIGURATION section of the `occnp_custom_values_24.3.0.yaml` file.

   The following is a sample Helm configuration. For more information on parameters and their supported values, see OAUTH Configuration.

   ```
   # ----OAUTH CONFIGURATION - BEGIN ----
     oauthValidatorEnabled: false
     nfInstanceId: 6faf1bbc-6e4a-4454-a507-a14ef8e1bc11
     allowedClockSkewSeconds: 0
     nrfPublicKeyKubeSecret: 'nrfpublickeysecret'
     nrfPublicKeyKubeNamespace: 'ocpcf'
     validationType: relaxed
     producerPlmnMNC: 123
     producerPlmnMCC: 456
     producerScope: nsmf-pdusession,nsmf-event-exposure
     nfType: PCF
     # ----OAUTH CONFIGURATION - END ----
   ```

Verifying oAuth Token

The following Curl command sends a request to create SM Policy with valid oAuth header:

```
curl --http2-prior-knowledge http://10.75.153.75:30545/npcf-
smpolicycontrol/v1/sm-policies -X POST -H 'Content-Type: application/json' -H
Authorization:'Bearer
eyJ0eXAiOiJKV1QiLCJraWQiOiI2MDFhZWQyYy1lMzE0LTQ2YTctYTNlNi1mMThjYTAyZmFheHgiLC
JhbGciOiJFUzI1NiJ9.eyJpc3MiOiI2NjRiMzQ0ZS03NDI5LTRjOGYtYTVkMi1lN2RmYWFhYmE0MDc
iLCJzdWIiOiJmZTdkOTkyYi0wNTQxLTRjN2QtYWI4NC1jNmQ3MGIxYjAxYjEiLCJhdWQiOiJTTUYiL
CJzY29wZSI6Im5zbWYtcGR1c2Vzc2lvbiIsImV4cCI6MTYxNzM1NzkzN30.oGAYtR3FnD33xOCmtUP
KBEA5RMTNvkfDqaK46ZEnnZvgN5Cyfgvlr85Zzdpo2lNISADBgDumD_m5xHJF8baNJQ' -d
'{"3gppPsDataOffStatus":true,"accNetChId":
{"accNetChaIdValue":"01020304","sessionChScope":true},"accessType":"3GPP_ACCES
S","dnn":"dnn1","gpsi":"msisdn-81000000002","ipv4Address":"192.168.10.10","ipv
6AddressPrefix":"2800:a00:cc01::/64","notificationUri":"http://
nf1stub.ocats.svc:8080/smf/
notify","offline":true,"online":false,"pduSessionId":1,"pduSessionType":"IPV4"
,"pei":"990000862471854","ratType":"NR","servingNetwork":
{"mcc":"450","mnc":"08"},"sliceInfo":
{"sd":"abc123","sst":11},"smPoliciesUpdateNotificationUrl":"npcf-
smpolicycontrol/v1/sm-policies/{ueId}/notify","subsSessAmbr":
{"downlink":"1000000 Kbps","uplink":"10000
Kbps"},"supi":"imsi-450081000000001","chargEntityAddr":
{"anChargIpv4Addr":"11.111.10.10"},"InterGrpIds":"group1","subsDefQos":
{"5qi":23,"arp":
{"priorityLevel":3,"preemptCap":"NOT_PREEMPT","preemptVuln":"NOT_PREEMPTABLE"}
,"priorityLevel":34},"numOfPackFilter":33,"chargingCharacteristics":"CHARGEING
","refQosIndication":true,"qosFlowUsage":"IMS_SIG","suppFeat":"","traceReq":
{"traceRef":"23322-
ae34a2","traceDepth":"MINIMUM","neTypeList":"32","eventList":"23","collectionE
ntityIpv4Addr":"12.33.22.11","collectionEntityIpv6Addr":"2001:db8:85a3::37:733
4","interfaceList":"e2"},"ueTimeZone":"+08:00","userLocationInfo":
{"nrLocation":{"ncgi":{"nrCellId":"51234a243","plmnId":
{"mcc":"450","mnc":"08"}},"tai":{"plmnId":
{"mcc":"450","mnc":"08"},"tac":"1801"}},"eutraLocation":{"tai":{"plmnId":
{"mnc":"08","mcc":"450"},"tac":"1801"},"ecgi":{"plmnId":
{"mnc":"08","mcc":"450"},"eutraCellId":"23458da"},"ageOfLocationInformation":2
33,"ueLocationTimestamp":"2019-03-13T06:44:14.34Z","geographicalInformation":"
AAD1234567890123","geodeticInformation":"AAD1234567890123BCEF","globalNgenbId"
:{"plmnId":{"mnc":"08","mcc":"450"},"n3IwfId":"n3iwfid"}},"n3gaLocation":
{"n3gppTai":{"plmnId":
{"mnc":"08","mcc":"450"},"tac":"1801"},"n3IwfId":"234","ueIpv4Addr":"11.1.100.
1","ueIpv6Addr":"2001:db8:85a3::370:7334","portNumber":30023}}}'
```

2.2.1.11 Configuring Policy to support Aspen Service Mesh

Policy leverages the Platform Service Mesh (for example, Aspen Service Mesh) for all internal and external TLS communication by deploying a special sidecar proxy in each pod to intercept all the network communications. The service mesh integration provides inter-NF communication and allows API gateway co-working with service mesh. The service mesh integration supports the services by deploying a special sidecar proxy in each pod to intercept all network communication between microservices.

Supported ASM versions: 1.11.x, and 1.14.6

For ASM installation and configuration, see official Aspen Service Mesh website for details.

The Aspen Service Mesh (ASM) configurations are categorized as follows:

- **Control Plane:** It involves adding labels or annotations to inject sidecar. The control plane configurations are part of the NF Helm chart.
- **Data Plane:** It helps in traffic management, such as handling NF call flows by adding Service Entries (SE), Destination Rules (DR), Envoy Filters (EF), and other resource changes like apiVersion change between versions. This configuration is done manually by using `occnp_custom_values_servicemesh_config_24.3.0.yaml` file .

Configuring ASM Data Plane

Data Plane configuration consists of the following Custom Resource Definitions (CRDs):

- Service Entry (SE)
- Destination Rule (DR)
- Envoy Filter (EF)
- Peer Authentication (PA)
- Authorization Policy (AP)
- Virtual Service (VS)
- RequestAuthentication

> **Note**
>
> Use `occnp_custom_values_servicemesh_config_24.3.0.yaml` Helm charts to add or delete CRDs that you may require due to ASM upgrades to configure features across different releases.

The Data Plane configuration is applicable in the following scenarios:

- **NF to NF Communication:** During NF to NF communication, where sidecar is injected on both NFs, SE and DR to communicate with the corresponding SE and DR of the other NF. Otherwise, sidecar rejects the communication. All egress communications of NFs must have a configured entry for SE and DR.

  > **Note**
  >
  > Configure the core DNS with the producer NF endpoint to enable the sidecar access for establishing communication between cluster.

- **Kube-api-server:** For Kube-api-server, there are a few NFs that require access to Kubernetes API server. The ASM proxy (mTLS enabled) may block this. As per F5 recommendation, the NF need to add SE for Kubernetes API server in its own namespace.
- **Envoy Filters:** Sidecars rewrite the header with its own default value. Therefore, the headers from back end services are lost. So, you need Envoy Filters to help in passing the headers from back end services to use it as it is.

The Custom Resources (CR) are customized in the following scenarios:

- **Service Entry:** Enables adding additional entries into Sidecar's internal service registry, so that auto-discovered services in the mesh can access or route to these manually specified

services. A service entry describes the properties of a service (DNS name, VIPs, ports, protocols, endpoints).

- **Destination Rule:** Defines policies that apply to traffic intended for service after routing has occurred. These rules specify configuration for load balancing, connection pool size from the sidecar, and outlier detection settings to detect and evict unhealthy hosts from the load balancing pool.
- **Envoy Filters:** Provides a mechanism to customize the Envoy configuration generated by Istio Pilot. Use Envoy Filter to modify values for certain fields, add specific filters, or even add entirely new listeners, clusters, and so on.
- **Peer Authentication:** Used for service-to-service authentication to verify the client making the connection.
- **Virtual Service:** Defines a set of traffic routing rules to apply when a host is addressed. Each routing rule defines matching criteria for the traffic of a specific protocol. If the traffic is matched, then it is sent to a named destination service (or subset or version of it) defined in the registry.
- **Request Authentication:** Used for end-user authentication to verify the credential attached to the request.
- **Policy Authorization:** Enables access control on workloads in the mesh. Policy Authorization supports `CUSTOM`, `DENY`, and `ALLOW` actions for access control. When `CUSTOM`, `DENY`, and `ALLOW` actions are used for a workload at the same time, the `CUSTOM` action is evaluated first, then the `DENY` action, and finally the `ALLOW` action.
  For more details on Istio Authorization Policy, see Istio / Authorization Policy.

Service Mesh Configuration File

A sample `occnp_custom_values_servicemesh_config_24.3.0.yaml` is available in **Custom_Templates** file. For downloading the file, see Customizing Policy.

> **Note**
>
> To connect to vDBTier, create an SE and DR for MySQL connectivity service if the database is in different cluster. Else, the sidecar rejects request as vDBTier does not support sidecars.

Table 2-9 Supported Fields in CRD

CRD	Supported Fields
Service Entry	• hosts • exportTo • addresses • ports.name • ports.number • ports.protocol • resolution

Table 2-9 (Cont.) Supported Fields in CRD

CRD	Supported Fields
Destination Rule	• host • mode • sbitimers • tcpConnectTimeout • tcpKeepAliveProbes • tcpKeepAliveTime • tcpKeepAliveInterval
Envoy Filters	• labelselector • applyTo • filtername • operation • typeconfig • configkey • configvalue • stream_idle_timeout • max_stream_duration • patchContext • networkFilter_listener_port • transport_socket_connect_timeout • filterChain_listener_port • route_idle_timeout • route_max_stream_duration • httpRoute_routeConfiguration_port • vhostname • cluster.service • type • listener_port • exactbalance
Peer Authentication	• labelselector • tlsmode
Virtual Service	• host • destinationhost • port • exportTo • retryon • attempts • timeout
Request Authentication	• labelselector • issuer • jwks/jwksUri
Policy Authorization	• labelselector • action • hosts • paths • xfccvalues

2.2.1.11.1 Predeployment Configurations

This sections explains the predeployment configuration procedure to install Policy with Service Mesh support.

Creating Policy namespace:

1. Verify required namespace already exists in system:

   ```
   kubectl get namespaces
   ```

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

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

   Where,

   `<namespace>` is the Policy namespace.

   For example:

   ```
   kubectl create namespace occnp
   ```

2.2.1.11.2 Installing Service Mesh Configuration Charts

Perform the below steps to configure Service Mesh CRs using the Service Mesh Configuration chart:

1. Download the service mesh chart `occnp_custom_values_servicemesh_config_24.3.0.yaml` file available in Custom_Templates directory.
   A sample `occnp_custom_values_servicemesh_config_24.3.0.yaml` is available in Custom_Templates file. For downloading the file, see Customizing Policy.

   > **Note**
   >
   > When Policy is deployed with ASM then cnDBTier is also installed in the same namespace or cluster, you can skip installing service entries and destination rules.

2. Configure the `occnp_custom_values_servicemesh_config_24.3.0.yaml` file as follows: Modify only the *"SERVICE-MESH Custom Resource Configuration" section* for configuring the CRs as needed. For example, to add or modify a ServiceEntry CR, required attributes and its value must be configured under the "serviceEntries:" section of "*SERVICE-MESH Custom Resource Configuration*". You can also comment on the CRs that you do not need.

   a. For updating **Service Entries**, make the required changes using the following sample template:

   ```
   serviceEntries:
     - hosts: |-
         [ "mysql-connectivity-
   service.<cndbtiernamespace>.svc.<clustername>" ]
       exportTo: |-
         [ "." ]
   ```

```
    location: MESH_EXTERNAL
    ports:
    - number: 3306
      name: mysql
      protocol: MySQL
    name: ocpcf-to-mysql-external-se-test
  - hosts: |-
      ".*<clustername>"
    exportTo: |-
      [ "." ]
    location: MESH_EXTERNAL
    ports:
    - number: 8090
      name: http2-8090
      protocol: TCP
    - number: 80
      name: HTTP2-80
      protocol: TCP
    name: ocpcf-to-other-nf-se-test
  - hosts: |-
      [ "kubernetes.default.svc.<clustername>" ]
    exportTo: |-
      [ "." ]
    location: MESH_INTERNAL
    addresses: |-
      [ "192.168.200.36" ]
    ports:
    - number: 443
      name: https
      protocol: HTTPS
    name: nf-to-kube-api-server
```

b. For customizing **Destination Rule**, make the required changes using the following sample template:

```
# destinationRules:
#  - host: "*.<clustername>"
#    mode: DISABLE
#    name: ocpcf-to-other-nf-dr-test
#    sbitimers: true
#    tcpConnectTimeout: "750ms"
#    tcpKeepAliveProbes: 3
#    tcpKeepAliveTime: "1500ms"
#    tcpKeepAliveInterval: "1s"
#  - host: mysql-connectivity-service.<clustername>.svc.cluster.local
#    mode: DISABLE
#    name: mysql-occne
#    sbitimers: false
```

c. For customizing **envoyFilters** according to the Istio version installed on the Bastion server, use any of the following templates:

For Istio version 1.11.x and 1.14.x

> **Note**
>
> Istio 1.11.x and 1.14.x support the same template for **envoyFilters** configurations.

```
envoyFilters_v_19x_111x:
  - name: set-xfcc-pcf
    labelselector: "app.kubernetes.io/instance: ocpcf"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnecti
onManager
        configkey: forward_client_cert_details
        configvalue: ALWAYS_FORWARD_ONLY
  - name: serverheaderfilter
    labelselector: "app.kubernetes.io/instance: ocpcf"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnecti
onManager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: custom-http-stream
    labelselector: "app.kubernetes.io/instance: ocpcf"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnecti
onManager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
        stream_idle_timeout: "6000ms"
        max_stream_duration: "7000ms"
        patchContext: SIDECAR_OUTBOUND
        networkFilter_listener_port: 8000
  - name: custom-tcpsocket-timeout
    labelselector: "app.kubernetes.io/instance: ocpcf"
    configpatch:
      - applyTo: FILTER_CHAIN
        patchContext: SIDECAR_INBOUND
        operation: MERGE
        transport_socket_connect_timeout: "750ms"
        filterChain_listener_port: 8000
  - name: custom-http-route
    labelselector: "app.kubernetes.io/instance: ocpcf"
```

```
    configpatch:
      - applyTo: HTTP_ROUTE
        patchContext: SIDECAR_OUTBOUND
        operation: MERGE
        route_idle_timeout: "6000ms"
        route_max_stream_duration: "7000ms"
        httpRoute_routeConfiguration_port: 8000
        vhostname: "ocpcf.svc.cluster:8000"
  - name: logicaldnscluster
    labelselector: "app.kubernetes.io/instance: ocpcf"
    configpatch:
      - applyTo: CLUSTER
        clusterservice: rchltxekvzwcamf-y-ec-
x-002.amf.5gc.mnc480.mcc311.3gppnetwork.org
        operation: MERGE
        logicaldns: LOGICAL_DNS
      - applyTo: CLUSTER
        clusterservice: rchltxekvzwcamd-y-ec-
x-002.amf.5gc.mnc480.mcc311.3gppnetwork.org
        operation: MERGE
        logicaldns: LOGICAL_DNS
```

> **Note**
>
> The parameter *vhostname* is mandatory when *applyTo* is **HTTP_ROUTE**

> **Note**
>
> Depending on the Istio version, update the correct value of envoy filters in the following line:`{{- range .Values.envoyFilters_v_19x_111x }}`

d. For customizing **PeerAuthentication**, make the required changes using the following sample template:

```
peerAuthentication:
   - name: default
     tlsmode: PERMISSIVE
   - name: cm-service
     labelselector: "app.kubernetes.io/name: cm-service"
     tlsmode: PERMISSIVE
   - name: ingress
     labelselector: "app.kubernetes.io/name: occnp-ingress-gateway"
     tlsmode: PERMISSIVE
   - name: diam-gw
     labelselector: "app.kubernetes.io/name: diam-gateway"
     tlsmode: PERMISSIVE
```

e. To customize the **Authorization Policy**, make the required changes using the following sample template:

```
#authorizationPolicies:
#- name: allow-all-provisioning-on-ingressgateway-ap
```

```
#  labelselector: "app.kubernetes.io/name: ingressgateway"
#  action: "ALLOW"
#  hosts:
#    - "*"
#  paths:
#    - "/nudr-dr-prov/*"
#    - "/nudr-dr-mgm/*"
#    - "/nudr-group-id-map-prov/*"
#    - "/slf-group-prov/*"
#- name: allow-all-sbi-on-ingressgateway-ap
#  labelselector: "app.kubernetes.io/name: ingressgateway"
#  action: "ALLOW"
#  hosts:
#    - "*"
# paths:
#   - "/npcf-smpolicycontrol/*"
#   - "/npcf-policyauthorization/*"
# xfccvalues:
#   - "*DNS=nrf1.site1.com"
#   - "*DNS=nrf2.site2.com"
#   - "*DNS=scp1.site1.com"
#   - "*DNS=scp1.site2.com"
#   - "*DNS=scp1.site3.com
```

f. **VirtualService** is required to configure the retry attempts for the destination host. For instance, for error response code value 503, the default behaviour of Istio is to retry two times. However, if the user wants to configure the number of retry attempts, then it can be done using `virtualService`.

   To customize the **VirtualService**, make the required changes using the following sample template:

   In the following example, the number of retry attempts are set to 0:

```
#virtualService:
#  - name: scp1site1vs
#    host: “scp1.site1.com”
#    destinationhost: “scp1.site1.com”
#    port: 8000
#    exportTo: |-
#      [ "." ]
#    attempts: "0"
#    timeout: 7s
#  - name: scp1site2vs
#    host: “scp1.site2.com”
#    destinationhost: “scp1.site2.com”
#    port: 8000
#    exportTo: |-
#      [ "." ]
#    retryon: 5xx
#    attempts: "1"
#    timeout: 7s
```

   Where, `host` or `destinationhost` value uses the format - <release_name>-<egress_svc_name>

To get the <egress_svc_name>, run the following command:

```
kubectl get svc -n <namespace>
```

For 5xx response codes, set the value of retry attempts to 1, as shown in the following sample:

```
#  - name: nrfvirtual2
#    host: ocpcf-occnp-egress-gateway
#    destinationhost: ocpcf-occnp-egress-gateway
#    port: 8000
#    exportTo: |-
#      [ "." ]
#    retryon: 5xx
#    attempts: "1"
```

g. **Request Authentication** is used to configure JWT tokens for Oauth validation. Network functions need to authenticate the OAuth token sent by consumer network functions by using the Public key of the NRF signing certificate and using service mesh to authenticate the token. Using the following sample format, users can configure **requestAuthentication** as per their system requirements:

   To customize the **Request Authentication**, make the required changes using the following sample template:

```
requestAuthentication:
#  - name: jwttokenwithjson
#    labelselector: httpbin
#    issuer: "jwtissue"
#    jwks: |-
#     '{
#       "keys": [{
#       "kid": "1",
#       "kty": "EC",
#       "crv": "P-256",
#       "x": "Qrl5t1-Apuj8uRI2o_BP9loqvaBnyM4OPTPAD_peDe4",
#       "y": "Y7vNMKGNAtlteMV-KJIaG-0UlCVRGFHtUVI8ZoXIzRY"
#      }]
#     }'
#  - name: jwttoken
#    labelselector: httpbin
#    issuer: "jwtissue"
#    jwksUri: https://example.com/.well-known/jwks.json
```

> **Note**
>
> For requestAuthetication, use either `jwks` or `jwksUri`.

3. Install the Service Mesh Configuration Chart as below:

- Run the below Helm install command on the namespace you want to apply the changes:

  ```
  helm install <helm-release-name> <charts> --namespace <namespace-name> -
  f <custom-values.yaml-filename>
  ```

  For example,

  ```
  helm install occnp-servicemesh-config occnp-servicemesh-
  config-24.3.0.tgz -n ocpcf -f
  occnp_custom_values_servicemesh_config_24.3.0.yaml
  ```

2.2.1.11.3 Deploying Policy with ASM

1. Create namespace label for auto sidecar injection to automatically add the sidecars in all of the pods spawned in Policy namespace:

   ```
   kubectl label --overwrite namespace <Namespace> istio-injection=enabled
   ```

   Where,

   `<Namespace>` is the Policy namespace.

   For example:

   ```
   kubectl label --overwrite namespace ocpcf istio-injection=enabled
   ```

2. The Operator should have special capabilities at service account level to start pre-install init container.
   Example of some special capabilities:

   ```
   readOnlyRootFilesystem: false
     allowPrivilegeEscalation: true
     allowedCapabilities:
     - NET_ADMIN
     - NET_RAW
     runAsUser:
       rule: RunAsAny
   ```

3. Customize the `occnp_custom_values_servicemesh_config_24.3.0.yaml` file for **ServiceEntries**, **DestinationRule**, **EnvoyFilters**, **PeerAuthentication**, **Virtual Service**, and **Request Authentication**.
4. Install Policy using updated `occnp_custom_values_servicemesh_config_24.3.0.yaml` file.

2.2.1.11.4 Postdeployment ASM configuration

This section explains the postdeployment configurations.

Run the below command to verify if all CRs are installed:

```
kubectl get <CRD-Name> -n <Namespace>
```

For example,

```
kubectl get
se,dr,peerauthentication,envoyfilter,vs,authorizationpolicy,requestauthenticat
ion -n ocpcf
```

Sample output for pods:

```
NAME
HOSTS                                                LOCATION
RESOLUTION   AGE
serviceentry.networking.istio.io/nf-to-kube-api-server
["kubernetes.default.svc.vega"]                      MESH_INTERNAL
NONE         17h
serviceentry.networking.istio.io/vega-ns1a-to-mysql-external-se-test
["mysql-connectivity-service.vega-ns1.svc.vega"]   MESH_EXTERNAL
NONE         17h
serviceentry.networking.istio.io/vega-ns1a-to-other-nf-se-test
["*.vega"]                                           MESH_EXTERNAL
NONE
17hNAME
HOST                                                    AGE
destinationrule.networking.istio.io/jaeger-dr                        occne-
tracer-jaeger-query.occne-infra                    17h
destinationrule.networking.istio.io/mysql-occne                      mysql-
connectivity-service.vega-ns1.svc.cluster.local   17h
destinationrule.networking.istio.io/prometheus-dr                    occne-
prometheus-server.occne-infra                      17h
destinationrule.networking.istio.io/vega-ns1a-to-other-nf-dr-test
*.vega
17hNAME                                                 MODE         AGE
peerauthentication.security.istio.io/cm-service     PERMISSIVE   17h
peerauthentication.security.istio.io/default        PERMISSIVE   17h
peerauthentication.security.istio.io/diam-gw        PERMISSIVE   17h
peerauthentication.security.istio.io/ingress        PERMISSIVE   17h
peerauthentication.security.istio.io/ocats-policy   PERMISSIVE
17hNAME                                                        AGE
envoyfilter.networking.istio.io/ocats-policy-xfcc           17h
envoyfilter.networking.istio.io/serverheaderfilter          17h
envoyfilter.networking.istio.io/serverheaderfilter-nf1stub   17h
envoyfilter.networking.istio.io/serverheaderfilter-nf2stub   17h
envoyfilter.networking.istio.io/set-xfcc-pcf
17hNAME                                              GATEWAYS
HOSTS                                  AGE
virtualservice.networking.istio.io/nrfvirtual1             ["vega-ns1a-occnp-
egress-gateway"]   17h
[cloud-user@vega-bastion-1 ~]$
```

Then, perform the steps described in Installing CNC Policy Package.

2.2.1.11.5 Disable ASM

This section describes the steps to delete ASM.

To Disable ASM, by running the following command:

```
kubectl label --overwrite namespace ocpcf istio-injection=disabled
```

where,

`namespace` is the deployment namespace used by helm command.

To see what namespaces have injection enabled or disabled, run the command:

```
kubectl get namespace -L istio-injection
```

In case, you want to uninstall ASM, disable ASM and then follow the below steps:

1. To Delete all the pods in the namespace:

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

2. To Delete ASM, run the following command:

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

   where,

   `<helm-release-name>` is the release name used by the `helm` command. This release name must be the same as the release name used for Service Mesh.

   `<namespace-name>` is the deployment namespace used by `helm` command

   For example:

   ```
   helm delete occnp-servicemesh-config -n ocpcf
   ```

3. To verify if ASM is disabled, run the following command:

   ```
   kubectl get se,dr,peerauthentication,envoyfilter,vs -n ocpcf
   ```

2.2.1.12 Anti-affinity Approach to Assign Pods to Nodes

Policy uses the anti-affinity approach to constrain a Pod to run only on the desired set of nodes. Using this approach, you can constrain Pods against labels on other Pods. It allows you to constrain on which nodes your Pods can be scheduled based on the labels of Pods already running on that node.

The following is a snippet for the anti-affinity specification:

```
affinity:
        podAntiAffinity:
          preferredDuringSchedulingIgnoredDuringExecution:
          - weight: 100
            podAffinityTerm:
              labelSelector:
                matchExpressions:
                - key: "app.kubernetes.io/name"
                  operator: In
```

```
            values:
            - {{ template "chart.fullname" .}}
          topologyKey: "kubernetes.io/hostname"
```

The description for the following parameters is as follow:

- `preferredDuringSchedulingIgnoredDuringExecution`: Specifies that the scheduler tries to find a node that meets the rule. If a matching node is not available, the scheduler still schedules the Pod.
- `weight`: For each instance of the `preferredDuringSchedulingIgnoredDuringExecution` affinity type, you can specify a weight between 1 and 100 (default).
- `matchExpressions`: The attributes under `matchExpressions` define the rules for constraining a Pod. Based on the preceding snippet, the scheduler avoids scheduling Pods having key as `app.kubernetes.io/name` and the value as `chart.fullname` on worker nodes having same value for label `kubernetes.io/hostname`. That is, avoid scheduling on Pod on same worker node when there are other worker nodes available with different value for label `kubernetes.io/hostname` and having no Pod with key as `app.kubernetes.io/name` and the value as `chart.fullname`.
- `topologyKey`: The key for the node label used to specify the domain.

  For anti-affinity approach to work effectively, every node in the cluster is required to have an appropriate label matching `topologyKey`. If the label is missing for any of the nodes, system may exhibit unintended behavior.

2.2.1.13 Configuring Network Policies

Network Policies allow you to define ingress or egress rules based on Kubernetes resources such as Pod, Namespace, IP, and Port. These rules are selected based on Kubernetes labels in the application. These Network Policies enforce access restrictions for all the applicable data flows except communication from Kubernetes node to pod for invoking container probe.

> **Note**
>
> Configuring Network Policy is optional. Based on the security requirements, Network Policy can be configured.

For more information on Network Policies, see https://kubernetes.io/docs/concepts/services-networking/network-policies/.

> **Note**
>
> - 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.

Configuring Network Policies

Network Policies support Container Network Interface (CNI) plugins for cluster networking.

> **Note**
>
> For any deployment with CNI, it must be ensured that Network Policy is supported.

Following are the various operations that can be performed for Network Policies:

2.2.1.13.1 Installing Network Policies

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 Policy. However, if Policy is already installed, you can still install the Network Policies.

To install Network Policies:

1. Open the `occnp-network-policy-custom-values.yaml` file provided in the release package zip file.
   For downloading the file, see Downloading Policy package and Pushing the Images to Customer Docker Registry.
2. The file is provided with the default Network Policies. If required, update the `occnp-network-policy-custom-values.yaml` file. For more information on the parameters, see Configuration Parameters for Network Policies.

> **Note**
>
> - To run ATS, uncomment the following policies from `occnp-network-policy-custom-values.yaml`:
>   - allow-egress-for-ats
>   - allow-ingress-to-ats
> - To connect with CNC Console, update the following parameter in the allow-ingress-from-console network policy in the `occnp-network-policy-custom-values.yaml`:
>
>   ```
>   kubernetes.io/metadata.name: <namespace in which CNCC is deployed>
>   ```
>
> - In `allow-ingress-prometheus` policy, `kubernetes.io/metadata.name` parameter must contain the value for the namespace where Prometheus is deployed, and `app.kubernetes.io/name` parameter value should match the label from Prometheus pod.

3. Run the following command to install the Network Policies:

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

   where:

   - <helm-release-name> is the occnp-network-policy helm release name.
   - <yaml-file> is the occnp-network-policy value file.
   - <namespace> is the OCCNP namespace.

   For example:

   ```
   helm install occnp-network-policy occnp-network-policy/ -n ocpcf -f occnp-network-policy-custom-values.yaml
   ```

> **Note**
>
> - 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 Policy and ATS in the same namespace.
> - It is highly recommended to run ATS after deploying Network Policies to detect any missing/invalid rule that can impact signaling flows.

2.2.1.13.2 Upgrading Network Policies

To add, delete, or update Network Policies:

1. Modify the `occnp-network-policy-custom-values.yaml` file to update, add, or delete the Network Policy.
2. Run the following command to upgrade the Network Policies:

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

   where:

   - <helm-release-name> is the occnp-network-policy helm release name.
   - <custom-value-file> is the occnp-network-policy value file.
   - <namespace> is the OCCNP namespace.

   For example:

   ```
   helm upgrade occnp-network-policy occnp-network-policy/ -n occnp
         -f occnp-network-policy-custom-values.yaml
   ```

2.2.1.13.3 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 occnp-network-policy -n occnp
```

Where,

- helm-release-name: occnp-network-policy Helm release name.
- namespace: CNC Console namespace.

2.2.1.13.4 Uninstalling Network Policies

Run the following command to uninstall network policies:

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

For example:

```
helm uninstall occnp-network-policy -n occnp
```

> **Note**
>
> While using the debug container, it is recommended to uninstall the network policies or update them as required to establish the connections.

2.2.1.13.5 Configuration Parameters for Network Policies

Table 2-10 Supported Kubernetes Resource for Configuring Network Policies

Parameter	Description	Details
apiVersion	This is a mandatory parameter. Specifies the Kubernetes version for access control. **Note**: This is the supported api version for network policy. This is a read-only parameter.	**Data Type**: string **Default Value**: `networking.k8s.i o/v1`
kind	This is a mandatory parameter. Represents the REST resource this object represents. **Note**: This is a read-only parameter.	Data Type: string Default Value: NetworkPolicy

Table 2-11 Supported Parameters for Configuring Network Policies

Parameter	Description	Details
metadata.name	This is a mandatory parameter. Specifies a unique name for Network Policies.	**DataType**: String **Default Value**: {{ .metadata.name }}
spec.{}	This is a mandatory parameter. This consists of all the information needed to define a particular network policy in the given namespace. **Note**: Policy supports the spec parameters defined in "Supported Kubernetes Resource for Configuring Network Policies".	**Default Value**: NA

For more information, see *Network Policies* in *Oracle Communications Cloud Native Core, Converged Policy User Guide*.

2.2.1.14 Configuring Traffic Segregation

This section provides information on how to configure Traffic Segregation in Policy. For description of " Traffic Segregation" feature, see " Traffic Segregation" section in "CNC Policy Features " chapter of *Oracle Communications Cloud Native Core, Converged Policy User Guide*.

Various networks can be created at the time of CNE cluster installation. The following things can be customized at the time of the cluster installation using `cnlb.ini` file provided as part of CNE installation.

1. Number of network pools
2. Number of Egress IPs
3. Number of Service IPs/Ingress IPs
4. External IPs/subnet

For more information, see *Oracle Communications Cloud Native Core, Cloud Native Environment User Guide*.

> **Note**
>
> - The network attachments will be deployed as a part of cluster installation only.
> - The network attachment name should be unique for all the pods.
> - The destination (egress) subnet addresses are known beforehand and defined under cnlb.ini file egress_dest variable to generate Network Attachment Definitions.

> **Note**
>
> When Policy is deployed with cnLB feature enabled, the TYPE field for the applicable Policy services shall remain to be "LoadBalancer" and the EXTERNAL-IP field shall be in pending state. This has no impact on the overall cnLB functionality in Policy application.

Configuration at Ingress Gateway

To use one or multiple interfaces, you must configure annotations in the `ingress-gateway.deployment.customExtension.annotations` parameter of the `occnp_custom_values_24.3.0_occnp.yaml` file.

```
ingress-gateway:
  deployment:
    customExtension:
      annotations: {
        # Enable this section for service-mesh based installation
#           traffic.sidecar.istio.io/excludeOutboundPorts: "9000,8095,8096",
#           traffic.sidecar.istio.io/excludeInboundPorts: "9000,8095,8096"
      }
```

Annotation for a single interface

```
k8s.v1.cni.cncf.io/networks: default/<network interface>@<network interface>,
oracle.com.cnc/cnlb: '[{"backendPortName": "<igw port name>", "cnlbIp":
"<external IP>","cnlbPort":"<port number>"}]'
```

Here,

- `k8s.v1.cni.cncf.io/networks`: Contains all the network attachment information the pod uses for network segregation.
- `oracle.com.cnc/cnlb`: To define service IP and port configurations that the deployment will employ for ingress load balancing.
  Where,
  - **cnlbIp** is the front-end IP utilized by the application.

- **cnlbPort** is the front-end port used in conjunction with the CNLB IP for load balancing.
- **backendPortName** is the backend port name of the container that needs load balancing, retrievable from the deployment or pod spec of the application.

> **Note**
>
> In case of TLS enabled for Ingress gateway, please use backendPortName as **igw-https**.

Sample annotation for a single interface:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig1-int8@nf-sig1-int8,
oracle.com.cnc/cnlb: '[{"backendPortName": "igw-http", "cnlbIp":
"10.123.155.16","cnlbPort":"80"}]'
```

Annotation for two or multiple interfaces

```
k8s.v1.cni.cncf.io/networks: default/<network interface1>@<network
interface1>, default/<network interface2>@<network interface2>,
oracle.com.cnc/cnlb: '[{"backendPortName": "<igw port name>", "cnlbIp":
"<network interface1>/<external IP1>, <network interface2>/<external
IP2>","cnlbPort":"<port number>"}]',
oracle.com.cnc/ingressMultiNetwork: "true"
```

Sample annotation for two or multiple interfaces:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig1-int8@nf-sig1-int8,default/nf-
sig2-int9@nf-sig2-int9,
oracle.com.cnc/cnlb: '[{"backendPortName": "igw-http", "cnlbIp": "nf-sig1-
int8/10.123.155.16,nf-sig2-int9/10.123.155.30","cnlbPort":"80"}]',
oracle.com.cnc/ingressMultiNetwork: "true"
```

Sample annotation for multiport:

```
k8s.v1.cni.cncf.io/networks: default/nf-oam-int5@nf-oam-int5,
oracle.com.cnc/cnlb: '[{"backendPortName": "query", "cnlbIp":
"10.75.180.128","cnlbPort": "80"},
{"backendPortName": "admin", "cnlbIp": "10.75.180.128", "cnlbPort":"16687"}]'
```

In the above example, each item in the list refers to a different backend port name with the same CNLB IP, but the ports for the front end are distinct.

Ensure that the backend port name aligns with the container port name specified in the deployment's specification, which needs to be load balanced from the port list. The CNLB IP represents the external IP of the service, and cnlbPort is the external-facing port:

```
ports:
- containerPort: 16686
  name: query
```

```
  protocol: TCP
- containerPort: 16687
  name: admin
  protocol: TCP
```

Configuration at Egress Gateway

To use one or multiple interfaces, you must configure annotations in the `egress-gateway.deployment.customExtension.annotations` parameter of the `occnp_custom_values_24.3.0_occnp.yaml` file.

```
egress-gateway:
  deployment:
    customExtension:
      annotations: {
        # Enable this section for service-mesh based installation
#           traffic.sidecar.istio.io/excludeOutboundPorts: "9000,8095,8096",
#           traffic.sidecar.istio.io/excludeInboundPorts: "9000,8095,8096"
      }
```

Sample annotation for a single interface:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig-egr1@nf-sig-egr1
```

Sample annotation for a multiple interface:

```
k8s.v1.cni.cncf.io/networks: default/nf-oam-egr1@nf-oam-egr1,default/nf-sig-egr1@nf-sig-egr1
```

Configuration at Diameter Gateway

Diameter gateway uses ingress-egress type of NAD to enable traffic flow for both ingress and egress directions.

To use one or multiple interfaces, you must configure annotations in the `diameter-gateway.deployment.customExtension.annotations` parameter of the `occnp_custom_values_24.3.0_occnp.yaml` file.

```
diameter-gateway:
  deployment:
    customExtension:
      annotations: {
        # Enable this section for service-mesh based installation
        # traffic.sidecar.istio.io/excludeOutboundPorts: "9000,5801",
        # traffic.sidecar.istio.io/excludeInboundPorts: "9000,5801"
      }
```

Annotation for a single interface:

```
k8s.v1.cni.cncf.io/networks: default/<network interface>@<network interface>,
oracle.com.cnc/cnlb: '[{"backendPortName":"diam-signaling", "cnlbIp":
"<externalIP>","cnlbPort":"<port number>"}]'
```

Here,

- **k8s.v1.cni.cncf.io/networks:** Contains all the network attachment information the pod uses for network segregation.
- **oracle.com.cnc/cnlb:** To define service IP and port configurations that the deployment will employ for diameter gateway ingress load balancing.
- **cnlbIp:** This is the front-end IP utilized by the application
- **cnlbPort:** This is the front-end port used in conjunction with the CNLB IP for load balancing.
- **backendPortName:** This is the backend port name of the container that needs load balancing, retrievable from the deployment or pod spec of the application.

> **Note**
>
> In case of TLS enabled for diameter gateway, please use backendPortName as **tls-signaling**.

Sample annotation for a single interface:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig1-ie1@nf-sig1-ie1,
oracle.com.cnc/cnlb:'[{"backendPortName":"diam-
signaling","cnlbIp":"10.123.155.17","cnlbPort":"3868"}]'
```

Annotation for two or multiple interfaces:

```
k8s.v1.cni.cncf.io/networks: default/<network interface1>@<network
interface1>, default/<networkinterface2>@<network interface2>,
oracle.com.cnc/cnlb: '[{"backendPortName":"diam-signaling","cnlbIp":
"<networkinterface1>/<external IP1>,<networkinterface2>/
<externalIP2>","cnlbPort":"<portnumber>"}]',
oracle.com.cnc/ingressMultiNetwork: "true"
```

Sample annotation for two or multiple interfaces:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig3-ie1@nf-sig3-ie1,default/nf-sig4-
ie1@nf-sig4-ie1,
oracle.com.cnc/cnlb:'[{"backendPortName":"diam-signaling","cnlbIp":"nf-sig3-
ie1/10.123.155.16,nf-sig4-ie1/10.123.155.30","cnlbPort":"3868"}]',
oracle.com.cnc/ingressMultiNetwork: "true"
```

Sample annotation for multiport:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig3-ie1@nf-sig3-ie1,
oracle.com.cnc/cnlb:'[{"backendPortName": "query", "cnlbIp":
"10.75.180.128","cnlbPort": "3868"},
{"backendPortName": "admin", "cnlbIp": "10.75.180.128","cnlbPort":"16687"}]'
```

In the above sample, each item in the list refers to a different backend port name with the same CNLB IP, but the ports for the front end are distinct.

Ensure that the backend port name aligns with the container port name specified in the deployment's specification, which needs to be load balanced from the port list. The CNLB IP represents the external IP of the service, and cnlbPort is the external-facing port:

```
    ports:
      -containerPort: 16686
        name: query
        protocol: TCP
      -containerPort: 16687
        name: admin
        protocol: TCP
```

Configuration at LDAP gateway

To use one or multiple interfaces, you must configure annotations in the `ldap-gateway.deployment.customExtension.annotations` parameter of the `occnp_custom_values_24.3.0_occnp.yaml` file.

```
ldap-gateway:
  deployment:
    customExtension:
      annotations: {
      }
```

Sample annotation for a single interface:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig-egr1@nf-sig-egr1
```

Sample annotation for a multiple interface:

```
k8s.v1.cni.cncf.io/networks:default/nf-oam-egr1@nf-oam-egr1,default/nf-sig-
egr1@nf-sig-egr1
```

Configuration at PRE Service

To use one or multiple interfaces, you must configure annotations in the `pre-service.deployment.customExtension.annotations` parameter of the `occnp_custom_values_24.3.0_occnp.yaml` file.

```
pre-service:
  deployment:
    customExtension:
      annotations: {
      }
```

Sample annotation for a single interface:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig-egr1@nf-sig-egr1
```

Sample annotation for a multiple interface:

```
k8s.v1.cni.cncf.io/networks:default/nf-oam-egr1@nf-oam-egr1,default/nf-sig-egr1@nf-sig-egr1
```

Configuration at notifier

To use one or multiple interfaces, you must configure annotations in the `notifier.deployment.customExtension.annotations` parameter of the `occnp_custom_values_24.3.0_occnp.yaml` file.

```
notifier:
  deployment:
    customExtension:
      annotations: {
      }
```

Sample annotation for a single interface:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig-egr1@nf-sig-egr1
```

Sample annotation for a multiple interface:

```
k8s.v1.cni.cncf.io/networks:default/nf-oam-egr1@nf-oam-egr1,default/nf-sig-egr1@nf-sig-egr1
```

For information about the above mentioned annotations, see "Configuring Cloud Native Load Balancer (CNLB)" in *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

2.2.2 Installation Tasks

This section provides the procedure to install Policy.

> **Note**
>
> - Before installing Policy, you must complete prerequiests and preinstallation tasks.
> - In a georedundant deployment, perform the steps explained in this section on all the georedundant sites.
> - In a Policy georedundant deployments, while adding a new Policy site ensure that its version is same as the other existing Policy site versions.

2.2.2.1 Installing Policy Package

This section describes the procedure to install Policy package.

To install the Policy package:

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

   ```
   cd occnp-<release_number>
   ```

2. Customize the `occnp_custom_values_occnp_24.3.0.yaml` or `occnp_custom_values_pcf_24.3.0.yaml` file (depending on the type of deployment model), with the required deployment parameters. See Customizing Policy chapter to customize the file.

   > **Note**
   >
   > - The parameters values mentioned in the custom-values.yaml file overrides the default values specified in the Helm chart. If the `envMyslqDatabase` parameter is modified, you must modify the `configDbName` parameter with the same value.
   > - The URL syntax for perf-info must be in the correct syntax otherwise, it keeps restarting. The following is a URL example for the bastion server if the BSF is deployed on OCCNE platform. On any other PaaS platform, the url should be updated according to the *Prometheus* and *Jaeger* query deployment.
   >
   >   ```
   >   # Values provided must match the Kubernetes environment.
   >   perf-info:
   >     configmapPerformance:
   >       prometheus: http://occne-prometheus-server.occne-infra.svc/
   >   clustername/prometheus
   >       jaeger:jaeger-agent.occne-infra
   >       jaeger_query_url:http://jaeger-query.occne-infra/clustername/
   >   jaeger
   >   ```
   >
   > - At least three configuration items must be present in the config map for perf-info, failing which perf-info will not work. If `jaeger` is not enabled, the `jaeger` and `jaeger_query_url` parameter can be omitted.

3. Run the following `helm install` commands:

- Install Policy using Helm:

  ```
  helm install -f <custom_file> <release_name> <helm-chart> --namespace
  <release_namespace> --atomic --timeout 10m
  ```

  For example:

  ```
  helm install -f occnp_custom_values_24.3.0.yaml occnp /home/cloud-user/
  occnp-24.3.0.tgz --namespace occnp --atomic
  ```

where:
helm_chart is the location of the Helm chart extracted from `occnp-24.3.0.tgz` file.

release_name is the release name used by helm command.

> **Note**
>
> *release_name* should not exceed the limit of 63 characters.

release_namespace is the deployment namespace used by helm command.

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

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*** (optional): If not specified, default value will be 300 (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 timeout value. Here, timeout value is not for overall install, but it is 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 `helm install` command. It leads to some anomalous behavior.

> **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
> ```

> **Note**
>
> The following warnings must be ignored for policy installation on 24.1.0, 24.2.0 and 24.3.0 CNE:
>
> ```
> helm install <release-name> -f <custom.yaml> <tgz-file> -n
> <namespace>
> W0311 11:38:44.824154  554744 warnings.go:70]
> spec.template.spec.containers[0].ports[4]: duplicate port
> definition with spec.template.spec.containers[0].ports[2]
> W0311 11:38:45.528363  554744 warnings.go:70]
> spec.template.spec.containers[0].ports[3]: duplicate port
> definition with spec.template.spec.containers[0].ports[2]
> W0311 11:38:45.684949  554744 warnings.go:70]
> spec.template.spec.containers[0].ports[4]: duplicate port
> definition with spec.template.spec.containers[0].ports[2]
> W0311 11:38:47.682599  554744 warnings.go:70]
> spec.template.spec.containers[0].ports[3]: duplicate port
> definition with spec.template.spec.containers[0].ports[1]
> W0909 12:21:54.735046 2509474 warnings.go:70]
> spec.template.spec.containers[0].env[32]: hides previous definition
> of "PRRO_JDBC_SERVERS", which may be dropped when using apply.
> NAME: <release-name>
> LAST DEPLOYED: <Date-Time>
> NAMESPACE: <namespace>
> STATUS: deployed
> REVISION: <N>
> ```

4. Press "Ctrl+C" to exit watch mode. We should run the `watch` command on another terminal. Run the following command to check the status:
   For Helm:

   ```
   helm status release_name
   ```

2.2.3 Postinstallation Task

This section explains the postinstallation tasks for Policy.

2.2.3.1 Verifying Policy Installation

To verify the installation:

1. Run the following command to verify the installation status:

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

   Where,

   `<release_name>` is the Helm release name of Policy.

   For example: `helm status occnp -n occnp`

   In the output, if `STATUS` is showing as `deployed`, then the deployment is succesful

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

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

   For example: `kubectl get pod -n occnp`

   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 occnp
   ```

2.2.3.2 Performing Helm Test

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

> **Note**
>
> - Helm test can be performed only on helm3.
> - If `nrf-client-nfmanagement.enablePDBSupport` is set to `true` in the custom-values.yaml, Helm test fails. It is an expected behavior as the mode is active and on standby, the leader pod (`nrf-client-management`) will be in ready state but the follower will not be in ready state, which will lead to failure in the Helm test.

Before running Helm test, complete the Helm test configurations under the Helm Test Global Parameters section in `custom-values.yaml` file. For more information on Helm test parameters, see *Global Parameters*.

Run the following command to perform the Helm test:

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

where:

`helm-release-name` is the release name.

`namespace` is the deployment namespace where Policy is installed.

Example:

```
helm test occnp -n occnp
```

Sample output:

```
Pod occnp-helm-test-test pending
Pod occnp-helm-test-test pending
```

```
Pod occnp-helm-test-test pending
Pod occnp-helm-test-test running
Pod occnp-helm-test-test succeeded
NAME: occnp-helm-test
LAST DEPLOYED: Thu May 19 12:22:20 2022
NAMESPACE: occnp-helm-test
STATUS: deployed
REVISION: 1
TEST SUITE:     occnp-helm-test-test
Last Started:   Thu May 19 12:24:23 2022
Last Completed: Thu May 19 12:24:35 2022
Phase:          Succeeded
```

If the Helm test failed, run the following command to view the logs:

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

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` model 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.
- If the Helm test fails, see *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.

2.2.3.3 Backing Up Important Files

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

- updated `occnp_custom_values_24.3.0.yaml`
- updated `occnp_custom_values_servicemesh_config_24.3.0.yaml` file
- updated helm charts
- secrets, certificates, and keys used during the installation

3
Customizing Policy

This chapter provides information about customizing Oracle Communications Cloud Native Core, Converged Policy (Policy) deployment in a cloud native environment.

The Policy deployment is customized by overriding the default values of various configurable parameters in the `occnp_custom_values_24.3.0.yaml` and `occnp_custom_values_pcf24.3.0.yaml` files.

> **Note**
>
> From release 22.2.x onwards, the `occnp-22.2.x-custom-values-pcrf.yaml` file is deprecated. To deploy Policy in PCRF mode, you must use the `occnp-22.2.x-custom-values-occnp.yaml` file.

To customize the custom yaml files, perform the following steps:

1. Unzip `Custom_Templates` file available in the extracted documentation release package. For more information on how to download the package from MOS, see Downloading Policy package section.
   The following files are used to customize the deployment parameters during installation:
   - `occnp_custom_values_24.3.0.yaml`: This file is used to customize the deployment parameters during Converged mode and PCRF mode deployment of Policy.
   - `occnp_custom_values_pcf24.3.0.yaml`: This file is used to customize the deployment parameters during PCF only mode deployment of Policy.
   - `occnp_custom_values_servicemesh_config_24.3.0.yaml`: This file is used while configuring ASM Data Plane.
2. Customize the appropriate custom value yaml file depending on the mode of deployment.
3. Customize the `occnp_custom_values_servicemesh_config_24.3.0.yaml` file, in case ASM Data Plane must be configured.
4. Save the updated files.

> **Note**
>
> - All parameters mentioned as mandatory must be present in `occnp_custom_values_24.3.0.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.

Customizing for PCRF Mode

This section provides information on how to use `occnp_custom_values_24.3.0.yaml` file for deploying Policy in PCRF mode. Users are required to enable only those services in the

custom yaml file that are required to run Policy in PCRF mode, and bring down other services down by setting their values to false in the custom yaml file.

The following table describes the services and their corresponding parameters that are required for deploying Policy in PCRF Mode:

Table 3-1 Enabling Policy Servicers

Service Name	Mandatory/Optional	Flag Name
AppInfo	Optional	appinfoServiceEnable
Bulwark Service	Optional	bulwarkServiceEnable
Notifier Service	Optional	notifierServiceEnable
Binding Service	Optional	bindingSvcEnabled
Diameter Connnector	Optional	diamConnectorEnable
Diameter Gateway	Optional	diamGatewayEnable
LDAP Gateway	Optional	ldapGatewayEnable
Alternate Route	Optional	alternateRouteServiceEnable
CHF Connector	Optional	chfConnectorEnable
Config Server	Mandatory	Enabled by default
Egress Gateway	Optional	NA
Ingress Gateway	Optional	NA
NRF Client-NF Discovery	Optional	nrfClientNfDiscoveryEnable
NRF Client-NF Management	Optional	nrfClientNfManagementEnable
UDR Connector	Optional	udrConnectorEnable
Audit Service	Mandatory	NA
CM Service	Mandatory	Enabled by default
PolicyDS	Mandatory	policydsEnable
PRE	Mandatory	Enabled by default
PRE Test	Optional	NA
Query Service	Mandatory	Enabled by default
AM Service	Optional	amServiceEnable
SM Service	Optional	smServiceEnable
UE Service	Optional	ueServiceEnable
PCRF-Core	Optional	pcrfCoreEnable
Perf Info	Optional	performanceServiceEnable
SOAP Connector	Optional	soapConnectorEnable
Usage Monitoring	Optional	usageMonEnable

3.1 Configurations for Pre and Post Upgrade/Install Validations

This section describes mandatory configurable parameters that you must customize in the `occnp_custom_values_24.3.0.yaml` file for successful validation checks required on the application, databases, and related tables before and after Policy application upgrade/install.

Table 3-2 Configuration Parameter for Pre and Post Flight Checks

Parameter	Description	Mandatory(M)/ Optional(O) Parameter	Accepted values	Default Value
global.hookValidation.dbSchemaValidate	Specifies to perform database validations in case of pre-installation, pre-upgrade/post-upgrade/post-installation. Checks if the required databases and tables exist. Validates that the required columns exist in the tables and the correct foreign key exists (for config-server).	M	true/false	false **Note**: By default, this flag is false. In that case, validations is performed, and if the validation fails, a warning is logged and install/upgrade will continue. If this flag is true and the validation fails, an error is thrown and installation/upgrade fails.
global.operationalState	Specifies to control deployment operationalState, mainly during fault recovery set up installation in inactive mode, i.e., complete shutdown mode.	M	• NORMAL • PARTIAL_SHUTDOWN • COMPLETE_SHUTDOWN	`&systemOperationalState NORMAL` **Note**: Need to use this field along with enabling the field `enableControlledShutdown` as `true`

Table 3-2 (Cont.) Configuration Parameter for Pre and Post Flight Checks

Parameter	Description	Mandatory(M)/ Optional(O) Parameter	Accepted values	Default Value
global.hookValidation.infraValidate	Specifies to perform pre-flight infrastructure related validations like Replication Status, Critical Alerts, Kubernetes Version, and cnDbtier Version. Infrastructure related validations are done in the very beginning of the upgrade/install and if it fails, then install/upgrade will fail at this stage.	M	true/false	false **Note**: • Ensure helm parameters for `replicationUri`, `dbTierVersionUri` and `alertmanagerUrl` are pointing to working URI/URL respectively. • Before enabling infra Validate flag, ensure that there are no critical alarms exists before upgrading/ installing a new release in order to avoid failures. Also, make sure that replication is up.
appinfo.dbTierVersionUri	Specifies the URI provided by the db monitor service to query the cnDBtier Version. For example: http://mysql-cluster-db-monitor-svc.occne-cndbtier:8080/db-tier/version	M	URI	Default Value is empty string: `""`.

Table 3-2 (Cont.) Configuration Parameter for Pre and Post Flight Checks

Parameter	Description	Mandatory(M)/ Optional(O) Parameter	Accepted values	Default Value
global.mySql.execution.ddlDelayTimeInMs	Adds a delay before the creation of configuration_item table, ensuring that topic_info table is created first and then the configuration_item table is created which has a foreign key dependency on topic_info. Specifies delay interval of 200 ms before inserting any entry into the ndb_replication table.	M	Interval in milliseconds	200 ms
appinfo.defaultReplicationStatusOnError	Specifies Replication Value in Case of any error on Infra Validation Replication Status	O	• UP • DOWN If the value is UP or empty string and the application throws an error while fetching replication status during infra-validation, the value of replication will be set as UP. If the value is DOWN, in case of any error while fetching replication status, the value of replication status will be set as DOWN.	UP
appinfo.nfReleaseVersion	Specifies the NF release version for the minViablePath validation.	O	NF release version If no value is provided, the minViablePath will validate the app-info-release version only.	Default Value is empty string: "".
global.alertmanagerUrl	Specifies the alertmanager POST uri, which will be used by the services to raise application level alerts.	O	URI	Default Value is empty string: "".

3.2 TLS Configurations

The following table describes the newly introduced and updated parameters for TLS:

Table 3-3 Helm Parameters

Parameter Name	Description	Mandatory/ Optional/ Conditional	Details
clientDisabledExtension	Disables the extension sent by messages originated by clients (ClientHello).	O	Data Type: String Range: NA Default Value: ec_point_formats
serverDisabledExtension	Disables the extension sent by messages originated by servers (ServerHello).	O	Data Type: String Range: NA Default Value: null
tlsNamedGroups	Provides a list of values sent in the supported_groups extension. These are comma-separated values.	O	Data Type: String Range: NA Default Value: null
clientSignatureSchemes	Provides a list of values sent in the signature_algorithms extension. These are comma-separated values.	O	Data Type: String Range: NA Default Value: null
tlsVersion	Indicates the TLS version.	M	Data Type: String Range: • TLSv1.2, TLSv1.3 • TLSv1.2 • TLSv1.3 Default Value: TLSv1.2, TLSv1.3
allowedCipherSuites	Indicates allowed Ciphers suites.	O	Data Type: String Range: NA Default Values: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • TLS_CHACHA20_POLY1305_SHA256 **Note:** Only the allowed cipher suite ciphers must be used in Cipher suite table.

Table 3-3 (Cont.) Helm Parameters

Parameter Name	Description	Mandatory/ Optional/ Conditional	Details
cipherSuites	Indicates supported cipher suites.	O	Data Type: String Range: NA Default Values: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • TLS_CHACHA20_POLY1305_SHA256

The following global sample Helm configuration is required for TLS 1.3:

```
global:
    tlsVersion: &tlsVersion 'TLSv1.2,TLSv1.3'
  # supportedCipherSuiteList: &supportedCipherSuiteList
'TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256'
  cipherSuites: &cipherSuites
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_AES_256_GCM_SHA384
    - TLS_AES_128_GCM_SHA256
    - TLS_CHACHA20_POLY1305_SHA256

  allowedCipherSuites: &allowedCipherSuites
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_AES_256_GCM_SHA384
    - TLS_AES_128_GCM_SHA256
    - TLS_CHACHA20_POLY1305_SHA256
```

The following Egress Gateway configuration is required for TLS 1.3:

```
egress-gateway:
#Cipher Suites to be enabled on client side
  clientDisabledExtension: null
  serverDisabledExtension: null
  tlsNamedGroups: null
  clientSignatureSchemes: null




#Enabling it for egress https requests
  enableOutgoingHttps: true

  #Enabling it for egress http1.1 requests
  http1:
    enableOutgoingHTTP1: false # Flag to enable or disable the feature

  egressGwCertReloadEnabled: false
  egressGwCertReloadPath: /egress-gw/store/reload

  service:
    ssl:
      tlsVersion: *tlsVersion
      privateKey:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        rsa:
          fileName: rsa_private_key_pkcs1.pem
        ecdsa:
          fileName: ssl_ecdsa_private_key.pem
      certificate:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        rsa:
          fileName: ocegress.cer
        ecdsa:
          fileName: ssl_ecdsa_certificate.crt
      caBundle:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        fileName: caroot.cer
      keyStorePassword:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        fileName: key.txt
      trustStorePassword:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        fileName: trust.txt



  httpsTargetOnly: "true"
```

```
  #true: Means change Scheme of RURI to http
  #false: Keep scheme as is.
  httpRuriOnly: "false"
```

> **Note**
>
> "httpsTargetOnly" must be set to `true` and "httpRuriOnly" must be set to `false`.

The following Ingress Gateway configuration is required for TLS 1.3:

```
ingress-gateway:
#Cipher Suites to be enabled on client side
  clientDisabledExtension: null
  serverDisabledExtension: null
  tlsNamedGroups: null
  clientSignatureSchemes: null


  # Enable it to accept incoming http requests
  enableIncomingHttp: false

  # ---- HTTPS Configuration - BEGIN ----
  enableIncomingHttps: true

  service:
    ssl:
      tlsVersion: *tlsVersion
      privateKey:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        rsa:
          fileName: rsa_private_key_pkcs1.pem
      certificate:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        rsa:
          fileName: ocegress.cer
      caBundle:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        fileName: caroot.cer
      keyStorePassword:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        fileName: key.txt
      trustStorePassword:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        fileName: trust.txt
```

For more information on HTTPS Configurations in Egress/Ingress Gateway, see Ingress/Egress Gateway HTTPS Configuration.

The following NRF configuration is required for TLS 1.3:

```
apiVersion: v1
data:
  profile: |-
    [appcfg]
    primaryNrfApiRoot=nf1stub.s-laplace.svc:8443
    nrfScheme=https
    retryAfterTime=PT120S
```

> **Note**
>
> NRF ports must be changed from `8080` to `8443`. Moreover, the `nrfScheme` must be changed from `http` to `https`.

3.3 TLS Configuration in Diameter Gateway

Configurable Parameters for TLS in Diameter Gateway:

Table 3-4 Configurable Parameters for TLS in Diameter Gateway

Parameter	Description	Mandatory/Optional/ Conditional	Default Vaue
TLS_ENABLED	To enable or disable TLS.	O	false
TLS_DIAMETER_PORT	Listening port diameter TLS.	O	5868
TLS_CIPHER_SUITE	To configure ciphers suites.	O	**TLS 1.2** • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 **TLS 1.3** • TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • TLS_CHACHA20_POLY1305_SHA256

Table 3-4 (Cont.) Configurable Parameters for TLS in Diameter Gateway

Parameter	Description	Mandatory/Optional/ Conditional	Default Vaue
TLS_INITIAL_ALGORITHM	To configure initial algorithm. ES256 or RS256	O	RS256
TLS_SECRET_NAME	Secret name for TLS configs	O	dgw-tls-secret
TLS_RSA_PRIVATE_KEY_FILENAME	To configure the filename for RSA private key, that will be stored in secret.	O	dgw-key.pem
TLS_ECDSA_PRIVATE_KEY_FILENAME	To configure the filename for ECDSA private key, that will be stored in secret.	O	dgw-ecdsa-private-key.pem
TLS_RSA_CERTIFICATE_FILENAME	To configure the filename for RSA certificate, that will be stored in secret.	O	dgw-cert.crt
TLS_ECDSA_CERTIFICATE_FILENAME	To configure the filename for ECDSA certificate, that will be stored in secret.	O	dgw-ecdsa-certificate.crt
TLS_CA_BUNDLE_FILENAME	To configure the filename for CA Bundle, that will be stored in secret.	O	ca-cert.cer
TLS_MTLS_ENABLED	To enable or disable mTLS	O	true

Note

Selective enabling of TLS version can be done through Diameter Gateway deployment file. The TLS_VERSION can be

- TLSv1.2, TLSv1.3
- TLSv1.2
- TLSv1.3

The following global sample Helm configuration is required for TLS in Diameter Gateway:

```
tls:
    enabled: false
    initialAlgorithm: 'RS256'
    secretName: 'dgw-tls-secret'
    cipherSuites:
      - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
      - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
      - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
```

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256
- TLS_CHACHA20_POLY1305_SHA256

3.4 Mandatory Configurations

This section describes the configuration parameters that are mandatory during the installation of Policy in any of the three supported modes of deployment.

To configure mandatory parameters, you should configure the following configurable parameters in the `occnp_custom_values_24.3.0.yaml` file:

Table 3-5 Configurable Parameters for Mandatory Configurations

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.nfInstanceId	Specifies the unique NF InstanceID for each site deployed for Policy. To setup georedundancy, users must specify the value while deploying Policy; otherwise, georedundancy will not be supported.	Yes	UUID **Note**: nfInstanceId should be provided as UUID for fresh installation.	Policy, PCF, & PCRF	Added in Release 1.10.0	On upgrade, the user should be using the original UUID or site Id which was provided during installation to avoid issues in upgrade. For upgrade, see Upgrading Policy. The same global `nfInstanceId` should be provided in appProfiles as well. The value of `nfInstanceId` must be unique for each site in a multi-site deployment.

Table 3-5 (Cont.) Configurable Parameters for Mandatory Configurations

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.dockerRegistry	Specifies the name of the Docker registry, which hosts Policy docker images.	Yes	Not applicable	Policy, PCF, & PCRF	Added in Release 1.0	This is a docker registry running OCCNE bastion server where all OAuth docker images are loaded. **Example** **occne-bastion:500** **occne-repo-host:5000**
global.envMysqlHost	Specifies the IP address or host name of the MySql server which hosts Policy database.	Yes	Not applicable	Policy, PCF, & PCRF	Added in Release 1.0	**Example** **10.196.33.106**
global.envMysqlPort	Specifies the MySql server port which hosts the Cloud Native Core Policy's databases.	Yes	Not applicable	Policy, PCF, & PCRF	Added in Release 1.0	**Example** **3306**
global.dbCredSecretName	Specifies the name of the Kubernetes secret object containing database username and password.	Yes	Not applicable	Policy, PCF, & PCRF	Added in Release 1.6.x	
global.privilegedDbCredSecretName	Specifies the name of the Kubernetes secret object containing database username and password for an admin user.	Yes	Not applicable	Policy, PCF, & PCRF	Added in Release 1.6.x	

Table 3-5 (Cont.) Configurable Parameters for Mandatory Configurations

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.releaseDbName	Specifies the name of the release database containing release version details.	Yes	Not applicable	Policy, PCF, & PCRF	Added in Release 1.6.x	

Here is a sample configuration for mandatory parameters in `occnp_custom_values_24.3.0.yaml` file:

```
global:
  # Docker registry name
  dockerRegistry: ''
  # Primary MYSQL Host IP or Hostname
  envMysqlHost: ''
  envMysqlPort: ''
  nrfClientDbName: 'occnp_nrf_client'
  nfInstanceId: "fe7d992b-0541-4c7d-ab84-c6d70b1b0123"
  # K8s secret object name containing OCPCF MYSQL UserName and Password
  dbCredSecretName: 'occnp-db-pass'
  privilegedDbCredSecretName: 'occnp-privileged-db-pass'
  #Release DB name containing release version details
  releaseDbName: 'occnp_release'
```

3.5 Enabling/Disabling Services Configurations

This section describes the configuration parameters that can be used to select the services that you want to enable or disable for your deployment.

To configure these parameters, you should configure the following configurable parameters in the `occnp_custom_values_24.3.0.yaml` file:

Table 3-6 Configurable Parameters for Enabling or Disabling the PCF Services

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.amServiceEnable	Specifies whether to enable or disable AM service.	No	True	• Converged Policy • PCF	Added in Release 1.7.1	If the user disables AM service by setting the value for this parameter as false, it is required to remove the AM service entry from `core-services.pcf` under `appinfo`.
global.smServiceEnable	Specifies whether to enable or disable SM service.	No	True	• Converged Policy • PCF	Added in Release 1.7.1	If the user disables SM service by setting the value for this parameter as false, it is required to remove the SM service entry from `core-services.pcf` under `appinfo`.
global.ueServiceEnable	Specifies whether to enable or disable UE service.	No	True	• Converged Policy • PCF	Added in Release 1.7.1	If the user disables UE service by setting the value for this parameter as false, it is required to remove the UE service entry from `core-services.pcf` under `appinfo`.

Table 3-7 Configurable Parameters for Enabling and Disabling the PCRF Core Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.pcrfCoreEnable	Specifies whether to enable or disable PCRF Core service.	No	True	• Converged Policy • cnPCRF	Added in Release 1.7.1

Table 3-8 Configurable Parameters for enabling or disabling Policy Data Source (PDS) Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.policydsEnable	Specifies whether to enable or disable Data Source service.	No	True	Policy, PCF, &cnPCRF	Added in Release 1.7.1	This parameter must be enabled when using LDAP, nUDR, and nCHF.
global.udrConnectorEnable	Specifies whether to enable or disable UDR connector.	No	True	Policy, PCF, &cnPCRF	Added in Release 1.9.0	Enable udr connector only when policyDS is enabled.
global.chfConnectorEnable	Specifies whether to enable or disable CHF connector.	No	True	Policy, PCF, &cnPCRF	Added in Release 1.9.0	Enable chf connector only when policyDS is enabled
global.ldapGatewayEnable	Specifies whether to enable or disable LDAP Gateway.	No	False	Policy, PCF, &cnPCRF	Added in Release 1.7.1	Applicable only when policy data source is LDAP server.
global.soapConnectorEnable	Specifies whether to enable or disable Soap connector.	No	False	Policy and PCRF	Added in Release 1.7.1	
global.userServiceEnable	Specifies whether to enable or disable User service.	No	false	Policy, PCF, and PCRF		Set the value for this parameter to true only when policyDS is disabled.

Table 3-8 (Cont.) Configurable Parameters for enabling or disabling Policy Data Source (PDS) Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
SM Data VSA Name	Indiates to provision subscriber name from Vendor Specific Attribute (VSA) data in SM Policy data for subscriber profile. **Example**: VendorSpecific-000111	No		Policy, PCF, and PCRF	Added in release 23.4.0.	

Table 3-9 Configurable Parameters for Enabling or Disabling the Audit Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
auditservice.enabled	Specifies whether to enable or disable Audit service.	No	true	Policy&PCF	Added in 1.7.1	
exceptionTableAuditEnabled	Specifies whether to enable or disable exception table audit.	No	false	Policy&PCF	Added in 23.4.0	Add this parameter to custom-values.yaml file for enabling the audit on exception tables.

Table 3-10 Configurable Parameters for Enabling or Disabling the Ingress and Egress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingressgateway.enabled	Specifies whether to enable or disable Ingress Gateway.	No	false	Policy, cnPCRF, &PCF	Added in Release 1.5.x	When depolyed in cnPCRF mode, enable this parameter only when soap connector is enabled.
egressgateway.enabled	Specifies whether to enable or disable Egress Gateway.	No	false	Policy &PCF	Added in Release 1.5.x	

> **Note**
>
> This section is applicable only when NF is required to register with NRF.

Table 3-11 Configurable Parameters for Enabling or Disabling the NRF Client Services

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.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.	No	true	Policy & PCF	Added in Release 1.7.1
global.nrfClientNfManagementEnable	Specifies whether to enable or disable NF Management service.	No	true	Policy & PCF	Added in Release 1.7.1
global.appinfoServiceEnable	Specifies whether to enable or disable app info service.	No	True	Policy & PCF	Added in Release 1.7.1
global.performanceServiceEnable	Specifies whether to enable or disable performance service.	No	True	Policy & PCF	Added in Release 1.7.1

Table 3-12 Configurable Parameters for Enabling/Disabling the Diamter Gateway/ Connector

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.diamConnectorEnable	Determines if the diameter connector is enabled or not.	No	True	Policy&PCF	Added in Release 1.7.1
global.diamGatewayEnable	Determines if the diameter gateway is enabled or not.	No	True	Policy, PCF, &cnPCRF	Added in Release 1.7.1

Table 3-13 Configurable Parameters for Enabling/Disabling the Binding Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.bindingSvcEnabled	Determines whether to enable or disable Binding service.	No	true	Policy	Updated in Release 1.14.0	The default value for this parameter is set to false in PCF and PCRF-Core custom values yaml files.

Table 3-14 Configurable Parameters for Enabling or Disabling the Bulwark Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.bulwarkServiceEnable	Determines whether to enable or disable the Bulwark service.	No	true	Policy and PCF	Added in Release 1.15.0

Table 3-15 Configurable Parameters for Enabling or Disabling the Notifier Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.notifierServiceEnable	Determines whether to enable or disable the Notifier service.	No	false	Policy and PCF	Added in Release 22.2.0

Table 3-16 Configurable Parameters for Enabling or Disabling the NWDAF Agent

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.nwdafAgentServiceEnable	Determines whether to enable or disable the NWDAF Agent.	No	false	Policy and PCF	Added in Release 22.4.0

Table 3-17 Configurable Parameters for Enabling or Disabling the Usage Monitoring Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.usageMonEnable	Determines whether to enable or disable the Usage Monitoring service.	No	false	Policy and PCF	Added in Release 22.2.0

Table 3-18 Configurable Parameters for Enabling/Disabling the Alternate Route Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.alternateRouteServiceEnable	Enable/Disable Alternate Route service	Yes	false	Policy & PCF	Added in Release 1.8.0	Enable this flag to include Alternate Route service as part of your Helm deployment.

Table 3-19 Configurable Parameters to enable or disable the resetContext flags for AM Service and UE Policy Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.resetContextUePolicySetData	The value of this flag is set to true if there are no existing UEPolicy Associations.	No	false	Policy & PCF	Added in Release 22.3.2	

Table 3-19 (Cont.) Configurable Parameters to enable or disable the resetContext flags for AM Service and UE Policy Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.resetContextAmPolicyData	The value of this flag is set to true if there are no existing AMService Associations.	No	false	Policy & PCF	Added in Release 22.3.2	
global.resetContextSsvOnAMCreate	If this flag is set to true, PDS SSV entry's context information is updated for AM context owner. If any AM context-info associated to SSV has exceeded the guard time, such context information is deleted.	No	false	Policy & PCF	Added in Release 23.1.0	This parameter is available in `values.yaml` file.
global.resetContextSSVOnUECreate	If this flag is set to true, PDS SSV entry's context information is updated for UE context owner. If any UE context-info associated to SSV has exceeded the guard time, such context information is deleted.	No	false	Policy & PCF	Added in Release 23.1.0	This parameter is available in `values.yaml` file.
global.enableSsvIdForReqParam	You can configure ENABLE_SSVID_FOR_REQPARAM for SM Service, AM Service, and UE Policy Service. When ENABLE_SSVID_FOR_REQPARAM flag is enabled, 'pdsSsvId' is added to the list of UserIds. When ENABLE_SSVID_FOR_REQPARAM flag is disabled, 'pdsSsvId' is not listed in the UserIds. PDS will work with old flow based on SUPI/ GPSI or PdsProfileId.	No	true	Policy & PCF	Added in Release 23.1.0	This parameter is available in `values.yaml` file.

The following is a sample configuration for configurable parameters related to service selection in the `occnp_custom_values_24.3.0.yaml` file used for deploying Policy:

```
global:
# Enable/disable PCF services
  amServiceEnable: true
  smServiceEnable: true
  ueServiceEnable: true
  nrfClientNfDiscoveryEnable: true
  nrfClientNfManagementEnable: true
  diamConnectorEnable: true
  appinfoServiceEnable: true
  performanceServiceEnable: true

# Enable userService only when policyDS is not enabled.
  userServiceEnable: false
  policydsEnable: true
# Enable udr and chf connectors only when policyDS is enabled
  udrConnectorEnable: true
  chfConnectorEnable: true
# Enable/disable PCRF services
  pcrfCoreEnable: true
  soapConnectorEnable: false

# Enable/disable common services
  bulwarkServiceEnable: true
  diamGatewayEnable: true
  bindingSvcEnabled: true

  ldapGatewayEnable: false
  alternateRouteServiceEnable: false



audit-service:
  enabled: false



ingress-gateway:
  enabled: false



egress-gateway:
  enabled: false
```

Table 3-20 Configurable Parameters to enable or disable the Swagger Console for UDR Connector and CHF Connector Services

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated /Updated in Release
global.swaggerUiEnabled	The value of this flag should be set to true to enable UDR Connector and CHF Connector services.	No	false	Policy & PCF	Added in Release 24.3.0

Configurable parameters to support binding header, routing binding header, and discovery header

Table 3-21 Configurable parameters to support binding header, routing binding header, and discovery header

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.SBI_BINDINGHEADER_SENDSCOPE	Enable/Disable scope in binding header.	No	true	Policy & PCF	Added in Release 23.2.4

3.6 Tracing Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` files to configure tracing.

Following are the common configurations for tracing:

Table 3-22 Common Configurable Parameters for Tracing

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
envJaegerCollectorHost	Specifies the host direction where the Jaeger Collector is found.	Mandatory	occne-tracer-jaeger-collector.occne-infra	CNC Policy, PCF, & PCRF	Added in Release 23.4.0	Make sure the jaeger Collector service is up and running inside OCCNE-Infra, with port specified in values.yaml

Table 3-22 (Cont.) Common Configurable Parameters for Tracing

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
envJaegerCollectorPort	Specifies the port where the Jaeger Collector is listening to receive spans.	Mandatory	4318	CNC Policy, PCF, & PCRF		Make sure this port matches with the one of your Jaeger Collector service port that is listening for OTLP formatted traces.
tracingEnabled	Specifies When 'true' enables the service to be instrumented by OpenTelemetry's Java Agent.	Mandatory	false	CNC Policy, PCF, & PCRF		
tracingSamplerRatio	Specifies a ratio of spans which will be sent to the Jaeger Collector; i.e. of the total amount of spans, specify how many are going to be sent to the Jaeger Collector.	Mandatory	.001	CNC Policy, PCF, & PCRF		Example: A value of "0.2" specifies that only 20 % of the spans are going to be sent. Range is 0 to 1.
tracingJdbcEnabled	Specifies when 'true' OpenTelemetry Java Agent will also show spans related to Database Operations.	Mandatory	false	CNC Policy, PCF, & PCRF		If tracingEnabled is true on deployment, this will be enabled by default. In case tracingEnabled is false, this will also be false by default
tracingLogsEnabled	Specifies when 'true' enables spans and tracing logging	Mandatory	false	CNC Policy, PCF, & PCRF		

Here is a sample configurations for tracing in `occnp_custom_values_24.3.0.yaml` file:

```
envJaegerCollectorHost: 'occne-tracer-jaeger-collector.occne-infra'
envJaegerCollectorPort: 4318 -> Make sure this matches with OCCNE-INFRA
jaeger collector service port.
tracing:
  tracingEnabled: 'true'
  tracingSamplerRatio: 0.001
  tracingJdbcEnabled: 'true'
  tracingLogsEnabled: 'false'
```

Note

These configurations are applicable to the following Policy services:

- Bulwark
- Binding Service
- Configuration Server
- PCRF core
- PRE
- LDAP Gateway
- Soap Connector
- CM Service
- Diameter Connector
- Query Service
- PCF AM Service
- PCF SM Service
- PCF UE Service
- PCF User-service
  - CHF Connector
  - UDR Connector
- PolicyDS
- Usage Monitoring

To configure tracing in ingress-gateway, you should configure the following configurable parameters in custom-value.yaml file:

Table 3-23 Configurable Parameters for Tracing Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.envJaegerAgentHost	Specifies the hostname or IP address for the jaeger agent	Yes	empty string	CNC Policy, PCF, & PCRF	Added in Release 1.0	This parameter is the FQDN of Jaeger Agent service running in OCCNE cluster under namespace occne-infra. Format is <JAEGER_SVC_NAME>.<JAEGER_NAMESPACE>
global.envJaegerQueryUrl	Specifies the query URL for the jaeger agent	Optional	empty string	CNC Policy, PCF, & PCRF	Added in Release 22.1.0	
ingress-gateway.jaegerTelemetryTracingEnabled	Specifies whether to enable or disable OpenTelemetry at Ingress Gateway.	No	false	CNC Policy, PCF, & PCRF	Added in Release 23.4.0	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values.

Table 3-23 (Cont.) Configurable Parameters for Tracing Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.openTelemetry.jaeger.httpExporter.host	Specifies the host name of Jaeger collector host	Yes, if `ingress-gateway.jaegerTelemetryTracingEnabled` flag is set to true	jaegercollector.cne-infra	CNC Policy, PCF, & PCRF		
ingress-gateway.openTelemetry.jaeger.httpExporter.port	Specifies the port of Jaeger collector port	Yes, if `ingress-gateway.jaegerTelemetryTracingEnabled` flag is set to true	4318	CNC Policy, PCF, & PCRF		

Table 3-23 (Cont.) Configurable Parameters for Tracing Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.openTelemetry.jaeger.probabilisticSampler	Specifies the sampler where value is between 0.0 (no sampling) and 1.0 (sampling of every request)	Yes, if `ingress-gateway.jaegerTelemetryTracingEnabled` flag is set to true	0.5	CNC Policy, PCF, & PCRF		The value range for Jaeger message sampler is 0 to 1. Value 0 indicates no Trace is sent to Jaeger collector. Value 0.3 indicates 30% of message is sampled and sent to Jaeger collector. Value 1 indicates 100% of message, that is, all the messages are sampled and sent to Jaeger collector.

Here is a sample configurations for tracing in ingress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
jaegerTelemetryTracingEnabled: *tracingEnabled

  openTelemetry:
    jaeger:
      httpExporter:
        host: *envJaegerCollectorHost
```

```
    port: *envJaegerCollectorPort
probabilisticSampler: *tracingSamplerRatio
```

Table 3-24 Configurable Parameters for Tracing Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.jaegerTelemetryTracingEnabled	Specifies whether to enable or disable Jaeger Tracing at Egress Gateway.	No	false	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	When this flag is set to true, make sure to update all Jaeger related attributes with the correct values.
egress-gateway.openTelemetry.jaeger.httpExporter.host	Specifies the host name of Jaeger collector host	Yes, if `egress-gateway.jaegerTelemetryTracingEnabled` flag is set to true.	jaegercollector.cne-infra	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
egress-gateway.openTelemetry.jaeger.httpExporter.port	Specifies the port of Jaeger collector port	Yes, if `egress-gateway.jaegerTelemetryTracingEnabled` flag is set to true.	4318	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	

Table 3-24 (Cont.) Configurable Parameters for Tracing Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.openTelemetry.jaeger.probabilisticSampler	Specifies the sampler where value is between 0.0 (no sampling) and 1.0 (sampling of every request)	Yes, if `egress-gateway.jaegerTelemetryTracingEnabled` flag is set to true.	0.5	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	The value range for Jaeger message sampler is 0 to 1. Value 0 indicates no Trace is sent to Jaeger collector. Value 0.3 indicates 30% of message is sampled and sent to Jaeger collector. Value 1 indicates 100% of message, that is, all the messages are sampled and sent to Jaeger collector.

Here is a sample configurations for tracing in egress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
jaegerTelemetryTracingEnabled: *tracingEnabled

  openTelemetry:
    jaeger:
      httpExporter:
        host: *envJaegerCollectorHost
```

```
    port: *envJaegerCollectorPort
  probabilisticSampler: *tracingSamplerRatio
```

To configure tracing in nrfClientNfDiscovery, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-25 Configurable Parameters for Tracing Configuration in nrfClientNfDiscovery

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client.nrf-client-nfdiscovery.envJaegerSamplerParam			'1'	CNC Policy & PCF	Added in Release 1.7.1	Applicable only when NRF Client services are enabled.
nrf-client.nrf-client-nfdiscovery.envJaegerSamplerType			ratelimitimg	CNC Policy & PCF	Added in Release 1.7.1	Applicable only when NRF Client services are enabled.
nrf-client.nrf-client-nfdiscovery.envJaegerServiceName			pcf-nrf-client-nfdiscovery	CNC Policy & PCF	Added in Release 1.7.1	Applicable only when NRF Client services are enabled.

Here is a sample configurations for tracing in `occnp_custom_values_24.3.0.yaml` file:

```
nrf-client-nfdiscovery:
    envJaegerSamplerParam: '1'
    envJaegerSamplerType: ratelimiting
    envJaegerServiceName: pcf-nrf-client-nfdiscovery
```

To configure tracing in nrfclientnfmanagement, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-26 Configurable Parameters for Tracing Configuration in nrfclientnfmanagement

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client.nrf-client-nfmanagement.envJaegerSamplerParam			'1'	CNC Policy & PCF	Added in Release 1.7.1.0	Applicable only when NRF Client services are enabled.
nrf-client.nrf-client-nfmanagement.envJaegerSamplerType			ratelimiting	CNC Policy & PCF	Added in Release 1.7.1	Applicable only when NRF Client services are enabled.
nrf-client.nrf-client-nfmanagement.envJaegerServiceName			pcf-nrf-client-nfmanagement	CNC Policy & PCF	Added in Release 1.7.1	Applicable only when NRF Client services are enabled.

Here is a sample configurations for tracing in `occnp_custom_values_24.3.0.yaml` file:

```
nrf-client-nfmanagement:
    envJaegerSamplerParam: '1'
    envJaegerSamplerType: ratelimiting
    envJaegerServiceName: pcf-nrf-client-nfmanagement
```

3.7 Database Name Configuration

This section describes the configuration parameters that can be used to customize the database names.

> **Note**
>
> Database name specified in the `occnp_custom_values_24.3.0.yaml` should be used while creating the database during installation. See Configuring Database, Creating Users, and Granting Permissions.

> **Note**
>
> The values of the parameters mentioned in the `occnp_custom_values_24.3.0.yaml` file overrides the default values specified in the helm chart. If the **envMysqlDatabase** parameter is modified, then you should modify the **configDbName** parameter with the same value.

Table 3-27 Customizable Parameters for Database Name Configuration for PCF Services

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
am-service.envMysqlDatabase	Name of the database for AM-Service	No	occnp_pcf_am	CNC Policy & PCF	Added in Release 1.0	Applicable only when AM service is enabled.
ue-service.envMysqlDatabase	Name of the database for UE-Service	No	occnp_pcf_ue	CNC Policy & PCF	Added in Release 1.0	Applicable only when UE service is enabled.
sm-service.envMysqlDatabase	Name of the database for SM-Service	No	occnp_pcf_sm	CNC Policy & PCF	Added in Release 1.0	Applicable only when SM service is enabled.
sm-service.envMysqlDatabaseUserService	Name of the database of User Service	No	occnp_pcf_user	CNC Policy & PCF	Deprecated in Release 1.10.0	Applicable only when SM service is enabled. Value of this parameter should be same as the value of "user-service.envMysqlDatabase" parameter.
config-server.envMysqlDatabase	Name of the database for Config Server service	No	occnp_config_server	CNC Policy & PCF	Added in Release 1.0	In case of Geo-redundancy, config-server database name for each site must be different.

Table 3-27 (Cont.) Customizable Parameters for Database Name Configuration for PCF Services

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
queryservice.envMysqlDatabaseSmService	Specify the database name of SM service	Conditional	occnp_pcf_sm	CNC Policy & PCF	Added in Release 1.6.x	Value of this parameter should be same as the value of "sm-service.envMysqlDatabase" parameter.

Table 3-28 Customizable Parameters for Database Name Configuration for Policy Data Source (PDS)

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
user-service.envMysqlDatabase	Name of the database for User-Service	No	occnp_pcf_user	CNC Policy & PCF	Deprecated in Release 1.10.0	Applicable only when user service is enabled.
policyds.envMysqlDatabase	Name of the database for Policy DS Service	No	occnp_policyds	CNC Policy, PCF, & PCRF	Added in Release 1.9.0	Applicable only when policyds is enabled.
policyds.envMysqlDatabaseConfigServer	Specify the database name of Config Server service.	No	occnp_config_server	CNC Policy, PCF, & PCRF	Added in Release 1.7.1	Applicable only when policyds is enabled.
policyds.envPdsDbMigrationFlag	It is recommended to keep the value as false for this parameter in multi-site deployment.	No	false	CNC Policy, PCF, & PCRF	Updated in Release 22.1.x	When rolling back to 1.15.x, ensure that the value of this parameter is false.

Table 3-29 Customizable Parameters for Database Name Configuration for PCRF Core Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
pcrf-core.envMysqlDatabase	Name of the database for PCRF-Core	No	occnp_pcrf_core	CNC Policy & cnPCRF	Added in Release 1.0	Applicable only when pcrf-core service is enabled.

Table 3-30 Customizable Parameters for Database Name Configuration for Binding Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
binding.envMysqlDatabase	Name of the database for Binding service	No	occnp_binding	CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	Applicable only when binding service is enabled.

Table 3-31 Customizable Parameters for Database Name Configuration for Audit Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
audit-service.envMysqlDatabase	Name of the database for Audit service	No	occnp_audit_service	CNC Policy & PCF	Added in Release 1.7.1	Applicable only when Audit service is enabled.

Table 3-32 Customizable Parameters for Database Name Configuration for CM Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
cm-service.envCommonConfigMysqlDatabase	Name of the database for CM service	No	occnp_commonconfig	CNC Policy, PCF, and PCRF	Added in Release 1.10.0	Applicable only when CM service is enabled.
cm-service.envMysqlDatabase	Name of the database for CM service.	No	occnp_cmservice	CNC Policy, PCF, and PCRF	Added in Release 1.15.0	Applicable only when CM service is enabled.
cm-service.envMysqlDatabaseConfigServer	Specify the database name of Config Server service.	No	occnp_config_server	CNC Policy, PCF, and PCRF	Added in Release 22.1.0	Applicable only when CM service is enabled.

Table 3-33 Customizable Parameters for Database Name Configuration for Notifier Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
notifier.envMysqlDatabaseConfigServer	Name of the database of Config Server for Notifier service.	No	occnp_config_server	CNC Policy & PCF	Added in Release 22.2.0	Applicable only when Notifier service is enabled.

Table 3-34 Customizable Parameters for Database Name Configuration for Usage Monitoring Service

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
usage-mon.envMysqlDatabase	Name of the database of Usage Monitoring service.	No	occnp_usagemon	CNC Policy, PCF & PCRF	Added in Release 22.2.0	Applicable only when Usage Monitoring service is enabled.

Here is a sample configuration for configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

```
am-service:
  envMysqlDatabase: occnp_pcf_am


sm-service:
  envMysqlDatabase: occnp_pcf_sm



config-server:
  envMysqlDatabase: occnp_config_server


queryservice:
  envMysqlDatabaseSmService: occnp_pcf_sm



audit-service:
  envMysqlDatabase: occnp_audit_service


policyds:
  envMysqlDatabase: 'occnp_policyds'
  envMysqlDatabaseConfigServer: 'occnp_config_server'


pcrf-core:
  # database name core service will connect to
  envMysqlDatabase: occnp_pcrf_core


binding:
  envMysqlDatabase: occnp_binding



ue-service:
  envMysqlDatabase: occnp_pcf_ue


cm-service:
  envCommonConfigMysqlDatabase: occnp_commonconfig
```

```
  envMysqlDatabase: occnp_cmservice
  envMysqlDatabaseConfigServer: 'occnp_config_server'


notifier:
  envMysqlDatabaseConfigServer: 'occnp_config_server'


usage-mon:
  envMysqlDatabase: occnp_usagemon
```

Cofiguring Database Engine

The following table describes the parameter that you can configure to customize the default database engine used by CNC Policy:

Table 3-35 Customizable Parameters for Database Engine for CNC Policy

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
dbConfig.dbEngine	Defines the MySQL engine that is used by CNC Policy to store information in the MySQL database.	Yes	NDBCLUSTER	CNC Policy, PCF, and PCRF	Added in Release 22.1.0.	If the database engine is not NDBCLUSTER, then the value for this parameter can be changed only during fresh installation of CNC Policy. Do not change the value of this parameter during upgrade scenarios.

Table 3-36 Customizable Parameters for Database Name Configuration for NRF Client

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.nrfClientDbName	Name of the database of NRF Client.	Yes	occnp_nrf_client	CNC Policy & PCF	Added in Release 23.4.0	Applicable for NRF Client.

Table 3-36 (Cont.) Customizable Parameters for Database Name Configuration for NRF Client

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfmanagement.dbConfig.leaderPodDbName	Name of the leader pod database for NRF Client.	Yes	occnp_leaderPodDb	CNC Policy & PCF		Applicable for NRF Client.

3.8 Database Load Balancing Configuration

This section describes the configurable parameters that can be used to configure connection load balancing across multiple MySQL nodes.

Table 3-37 Configurable Parameters for Database Load Balancing Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.envMysqlLoadBalancingEnabled	Specifies if the load balancing is enabled or disabled among all MySQL nodes.	No	false	CNC Policy, PCF, &cnPCRF	Updated in Release 1.10.4	Applicable only to AM, SM, UE and PolicyDS services. It is recommended to set its value to true when MySQL connectivity with headless service from occne is used to connect with external database.

Table 3-37 (Cont.) Configurable Parameters for Database Load Balancing Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.envMysqlDnsSrvEnabled	Specifies if services use DNS SRV records for connecting to MySQL servers.	No	false	CNC Policy, PCF, &cnPCRF	Added in 1.10.0	Applicable only to AM, SM, UE and PolicyDS services. It is recommended to set its value to true when MySQL connectivity with headless service from occne is used to connect with external database.
global.envMysqlLoadBalanceHosts	Distributes read and/or write load across multiple MySQL server instances for Cluster. Users can configure it in the following two ways: 1. list of mysql nodes in comma separated list format, as shown below: [_host_1][:_port_],[_host_2][:_port_] **Example**: 10.75.152.89:3306,10.75.152.86:3306 2. mysql service name to load-balance by making use of DNS SRV records **Example**: mysql-connectivity-service-headless.occne-infra.svc.policy-bastion Note: For this method, make sure that `global.envMysqlDnsSrvEnabled` parameter is set to true.	No	NA	CNC Policy, PCF, &cnPCRF	Added in Release 1.10.4	Configure this parameter only when `global.envMysqlLoadBalancingEnabled` parameter is set to true.

Table 3-38 Configurable Parameters for Service Specific Table Slicing

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
usage-mon.enableTableSlicing	Indicates whether to slice the UmContext table for Usage Monitoring into different slices so that any create/retrieve/update/ insert/delete operation on this table is performed on the sliced tables rather than the main table.	Optional	false	PCRF Core	Policy 24.1.0
usage-mon.umContextTableSlicingCount	Specifies the number of slices to be created for UmContext table.	Optional	1	PCRF Core	Policy 24.1.0
enableTableSlicing	Indicates whether to slice the SMPolicyAssociation table for SM Service into different slices so that any create/ retrieve/update/insert/delete operation on this table is performed on the sliced tables rather than the main table.	Optional	false	PCF	
smPolicyAssociationTableSlicingCount	Specifies the number of slices to be created for SMPolicyAssociation table.	Optional	1	PCF	
GX_SESSION_TABLE_SLICING_ENABLED	Indicates whether to slice the GxSession table for PCRF Core service into different slices so that any create/retrieve/update/ insert/delete operation on this table is performed on the sliced tables rather than the main table.	Optional	false	PCRF Core	Policy 24.2.0
GX_SESSION_TABLE_SLICING_COUNT	Specifies the number of slices to be created for GxSession table.	Optional	1	PCRF Core	Policy 24.2.0

> **Note**
>
> The above mentioned parameters listed under *Configurable Parameters for Database Slicing* section are not available in `custom-values.yaml` file. Contact Oracle Customer Support team to configure the database slicing for any of the Policy microservices.

3.9 Database Connection Timers Configuration

This section describes the configurable parameters that can be used to customize the database connection timers.

> **Note**
>
> In this release, the parameters described in this section are applicable to only SM service and PolicyDS.

Table 3-39 Customizable Parameters for Database Connection Timers Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.mySql.connection.maxLifeTime	Specifies the maximum lifetime (in milliseconds) of a connection.	No	540000	CNC Policy & PCF	Added in Release 1.10.4	
global.mySql.connection.idleTimeout	Specifies the maximum amount of time (in milliseconds) that a connection can remain idle. On the expiry of idle timer, the connection shall be closed.	No	540000	CNC Policy & PCF	Added in Release 1.10.4	
global.mySql.connection.connectionTimeout	Specifies the maximum number of milliseconds the application shall wait to get a connection from pool.	No	2000	CNC Policy & PCF	Added in Release 1.10.4	
global.mySql.connection.validationTimeout	Specifies the maximum number of milliseconds that the application shall wait for a connection to be validated as alive	No	500	CNC Policy & PCF	Added in Release 1.10.4	
global.mySql.connection.socketTimeout	Specifies the timout (in milliseconds) on network socket operations for a database connection.	No	3000	CNC Policy & PCF	Added in Release 1.10.4	
global.mySql.loadBalance.serverBlocklistTimeout	Specifies the time (in milliseconds) between two consecutive checks on servers which are unavailable, by controlling how long a server lives in the global blocklist.	No	60000	CNC Policy PCF PCRF Core	Added in Release 1.11.1	Configure this parameter when `global.envMysqlLoadBalancingEnabled` is set to true. This parameter is applicable to only PolicyDS.

Here is a sample configuration for configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

```
mySql:
    connection:
      maxLifeTime: '540000'
      idleTimeout: '540000'
      connectionTimeout: '2000'
      validationTimeout: '500'
      socketTimeout: '3000'
    loadBalance:
      serverBlocklistTimeout: '60000'
```

This section describes the configurable parameters that can be used to resolve the database conflict.

> **Note**
>
> These configurations are only available if the database is MySQL cluster (NDB).

Table 3-40 Configurable Parameters to enable or disable the Conflict Resolution

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.mySql.conflictResolution.ndbConflictResolutionEnabled	This flag is used to prevent data conflicts in georeplicated deployments. When there are multiple sites with real-time replication, if a session is updated at both sites simultaneously, this is considered as a conflict. This flag configures the MySQL cluster replication to compare the updated timestamp in the session record, so the conflicts can be automatically resolved.	No	True	CNC Policy and PCF	Added in Release 1.12.0	This feature is only available if the database is MySQL cluster (NDB). For MySQL (innodb), the value for this flag must be set to false. **Note**: Even if its a single-site cnPolicy NF deployment, set this parameter to true. As this will keep georedundancy and geo-replication enabled among the sites during multi-site deployment.
global.mySql.conflictResolution.useMaxDeleteWinInsConflictFn	This flag is used to update the Conflict Resolution Function to MAX_DEL_WIN_INS.	No	True	CNC Policy and PCF	Added in Release 22.4.0	This feature is available if the NDB version is 8.0.30. If NDB version is less than 8.0.30, the value for this flag must be set to false.

Here is a sample configuration for configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

```
global:
    mySql:
      conflictResolution:
        ndbConflictResolutionEnabled: true
        useMaxDeleteWinInsConflictFn: true
```

3.10 Configurations for DB Compression

3.10.1 PCRF-Core

The corresponding configurations are detailed as follows:

> **Important**
>
> You must consult the My Oracle Support (https://support.oracle.com) to enable or disable the application-based DB compression.

Table 3-41 DB Compression Configurations

Name	Default Value	custom.yaml Configurable	Helm Configurable	Advanced Settings Configurable	Description
DB_COMPRESSION_MYSQL_ENABLED	false	mySqlDbCompressionEnabled: 'false'	Yes	No	Enables or disables MySQL based data compression for 'value' column in the gxsession, rxsession, and sdsession tables in pcrf-core. Possible values: 'true', 'false'.
DB_COMPRESSION_MYSQL_COMPRESSIONSCHEME	0	mySqlDbCompressionScheme: '0'	Yes	No	For a record inserted or updated in pcrf-core's gxsession, rxsession and/or sdsession table, a column named 'compression_scheme' in those tables will reflect this (0/1) value. Possible values: '0': represents DISABLED '1': represents ZLIB_COMPRESSION_MYSQL

Table 3-42 Miscellaneous Configurations

Name	Default Value	custom.yaml Configurable	Helm Configurable	Advanced Settings Configurable	Description
DIAMETER_MSG_BUFFER_THREAD_COUNT	60	diameterMsgBufferThreadCount: 60	Yes	No	The number of threads that will be used to process read Diameter messages and process to completion. If this is set to 0, then the MsgBuffer will not be used, and the ReadThreads will process the message to completion. Using this thread pool gives you reduced latency at the expense of throughput. **Note:** It is recommended not to change this value without consulting My Oracle Support (https://support.oracle.com), as optimal value for this configuration depends on many factors.
DIAMETER_MSG_BUFFER_QUEUE_SIZE	8192	diameterMsgBufferQueueSize: 8192	Yes	No	The size of the queue holding pending messages which have been readoff the socket, but not yet processed. **Note:** It is recommended not to change this value without consulting My Oracle Support (https://support.oracle.com), as optimal value for this configuration depends on many factors.
ADMISSION_DIAMETER_REQUEST PROCESSINGLIMIT	5000	diameterRequestProcessingLimit: '5000'	Yes	No	Specifies the maximum amount of time, in milliseconds, a request can be processed before being dropped, if no answer has been sent. *Possible values*: The value of this key can be less than or equal to "Response Timeout (sec)" configuration in Policy.
PRRO_JDBC_QUERY_TIMEOUT	2000	envDbQueryTimeout: 2000	Yes	No	Specifies the timeout on JDBC statements, *in milliseconds*. When timeouts are set, the driver would wait for the given number of seconds for the query to execute and throw an SQLTimeoutException if it does not respond within that time.

3.10.2 SM Service

This section describes the customizatons that you should make in custom-value.yaml files to configure DB Compression in SM Service.

To configure DB Compression in SM Service, you should configure the following configurable parameter in custom-value.yaml file:

Table 3-43 Configurable Parameters for DB Compression in SM Service

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
smDataCompressionScheme	Specifies the control of "Data Compression Scheme" configuration in SM Service during install or upgrade. Possible values: 0, 1, or 2.	Optional	0	CNC Policy& PCF	23.2.0

3.10.3 PA Service

This section describes the customizatons that you should make in custom-value.yaml files to configure DB Compression in PA Service.

To configure DB Compression in PA Service, you should configure the following configurable parameter in custom-value.yaml file:

Table 3-44 Configurable Parameters for DB Compression in PA Service

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
paDataCompressionScheme	Specifies the control of "Data Compression Scheme" configuration in PA Service during install or upgrade. Possible values: 0, 1, or 2.	Optional	0	CNC Policy& PCF	23.2.0

3.11 NRF Client Configuration

This section describes the NRF Client configuration parameters.

> **Note**
>
> These configurations are required when CNC Policy is required to register with NRF. Before configuring NRF client configuration, you must enable NRF Client services.

To configure these parameters, you should configure the following configurable parameters in the `occnp_custom_values_24.3.0.yaml` file:

Table 3-45 Global Configurable Parameters for NRF Client Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.nrfClientDbName	Specifies the occnp_nrf_client database name in the global parameters.	Yes	occnp_nrf_client	CNC Policy & PCF	Added in 23.4.0
global.deploymentNrfClientService.envNfNamespace	Specifies the Kubernetes namespace of Policy.	Yes	Not Applicable	CNC Policy & PCF	Added in Release 1.6.x
global.nrfClientCommonServicePort	Specifies the port to be used for readiness and liveness probes.	Yes	9000 **Note**: Changing this value may result in errors, when starting Nrf-Client pods.	CNC Policy & PCF	Added in Release 24.1.0

Here is a sample configuration for NRF client customization parameters under **global** section in `occnp_custom_values_24.3.0.yaml`

```
global:
  nrfClientDbName: 'occnp_nrf_client'

  deploymentNrfClientService:
    #K8s namespace of PCF
    envNfNamespace: ''

  nrfClientCommonServicePort: *containerMonitoringHttp
```

Table 3-46 Configurable Parameters for NRF Client Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client.configmapApplicationConfig	This config map is used to provide inputs to NRF-Client.	Yes	Not Applicable	CNC Policy & PCF		
&configRef	This reference variable is used to take the input from the config map.	Yes	Not Applicable	CNC & Policy	Added in Release 1.14.0	Users must not make any alterations to this variable.

Table 3-46 (Cont.) Configurable Parameters for NRF Client Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client.configmapApplicationConfig.profile	Contains configuration parameters that goes into nrf-client's config map	Yes	Not Applicable	CNC Policy & PCF	Added in Release 1.6.x	Refer config-map table for configurable parameters.

Table 3-46 (Cont.) Configurable Parameters for NRF Client Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
appinfo.infraServices	Specifies the URI for the health check of InfraServices that need to be monitored. **Examples**: http://mysql-cluster-db-monitor-svc.vzw1-cndbtier:8080/actuator/health http://mysql-cluster-db-replication-svc.vzw1-cndbtier/actuator/health Uncomment this parameter and set this parameter to an empty array if any one of following conditions is true: • Deploying on OCCNE 1.4 or lower version • Not deploying on OCCNE • Do not wish to monitor infra services such as db-monitor service	Conditional	Not Applicable	CNC Policy & PCF	Added in Release 1.7.1	This parameter uses the default namespace - `occne-infra`. If cnDBTier is used to deploy CNC Policy, this field must be updated accordingly.

Table 3-46 (Cont.) Configurable Parameters for NRF Client Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
appinfo.core_services.pcf	Specifies the list of PCF services to be monitored.	Optional	- '{{ template "service-name-pcf-sm" . }}' - '{{ template "service-name-pcf-am" . }}' - '{{ template "service-name-pcf-ue" . }}'	CNC Policy & PCF	Added in Release 1.14.0	
appinfo.core_services.common	Specifies the list of common services to be monitored.	Optional	- '{{ template "service-name-ingress-gateway" . }}' - '{{ template "service-name-oc-diam-gateway" . }}' - '{{ template "service-name-nrf-client-nfmanagement" . }}'	CNC Policy & PCF	Added in Release 1.14.0	
perf-info.configmapPerformance.prometheus	Specifies Prometheus server URL	Conditional	http://occne-prometheus-server.occne-infra	CNC Policy & PCF	Added in Release 1.0	If no value is specified, PCF reported 0 loads to NRF.

Table 3-46 (Cont.) Configurable Parameters for NRF Client Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
notifySemanticValidationEnabled	Specifies wether to enable or disable the NFProfile validations.	Mandatory	True	CNC Policy & PCF	Added in Release 23.2.0	NA

> **Note**
>
> For `perf-info.configmapPerformance.prometheus` parameter, you must provide URL in proper format, along with at least three configuration items. If any of the configuration items, as shown in the following sample code, is not provided perf-info service may not work. If jaeger is not enabled, the jaeger and jaeger_query_url parameter can be omitted. The sample values must be updated to match the Kubernetes environment.
>
> ```
> perf-info:
>   serviceMeshCheck: *serviceMeshEnabled
>   istioSidecarReadyUrl: *istioSidecarReadyUrl
>   istioSidecarQuitUrl: *istioSidecarQuitUrl
>   configmapPerformance:
>     prometheus: http://occne-prometheus-server.occne-infra.svc
>     jaeger=jaeger-agent.occne-infra
>     jaeger_query_url=http://jaeger-query.occne-infra
> ```

Configurable parameters NRF Client Configuration

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
configmapApplicationConfig.profile.primaryNrfApiRoot	Primary NRF hostname and port <Hostname/IP>:<Port>	valid api root	CNC Policy & PCF	Added in Release 1.6.x	For Example: nrf1-api-gateway.svc:80
configmapApplicationConfig.profile.SecondaryNrfApiRoot	secondary NRF hostname and port <Hostname/IP>:<Port>	valid api root	CNC Policy & PCF	Added in Release 1.6.x	For Example: nrf2-api-gateway.svc:80

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
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.	valid ISO 8601 duration format	CNC Policy & PCF	Added in Release 1.6.x	For Example: PT120S
configmapApplicationConfig.profile.nrfClientType	The NfType of the NF registering. This should be set to PCF.	PCF	CNC Policy & PCF	Added in Release 1.6.x	
configmapApplicationConfig.profile.nrfClientSubscribeTypes	NF Type(s) for which the NF wants to discover and subscribe to the NRF.	BSF,UDR,CHF	CNC Policy & PCF	Added in Release 1.6.x	Leave blank if PCF does not require.
configmapApplicationConfig.profile.appProfiles	NfProfile of PCF to be registered with NRF.	Valid NF Profile	CNC Policy & PCF	Added in Release 1.6.x	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*. During fresh install the value of this parameter is loaded into the database and then used to trigger NfRegister or NfUpdate operation to NRF. For any subsequent changes to appProfile, REST API or CNC Console must be used. For more information, see *Oracle Communications Cloud Native Core Policy REST Specification Guide* or *Oracle Communications Cloud Native Core Policy User Guide.*
configmapApplicationConfig.profile.enableF3	Support for 29.510 Release 15.3	true/false	CNC Policy & PCF	Added in Release 1.6.x	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
configmapApplicationConfig.profile.enableF5	Support for 29.510 Release 15.5	true/false	CNC Policy & PCF	Added in Release 1.6.x	
configmapApplicationConfig.profile.renewalTimeBeforeExpiry	Time Period(seconds) before the Subscription Validity time expires	Time in seconds	CNC Policy & PCF	Added in Release 1.6.x	For Example: 3600 (1hr)
configmapApplicationConfig.profile.validityTime	The default validity time(days) for subscriptions	Time in days	CNC Policy & PCF	Added in Release 1.6.x	For Example: 30 (30 days)
configmapApplicationConfig.profile.enableSubscriptionAutoRenewal	Enable Renewal of Subscriptions automatically	true/false	CNC Policy & PCF	Added in Release 1.6.x	
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	80	CNC Policy & PCF	Added in Release 1.14.0	
configmapApplicationConfig.profile.acceptAdditionalAttributes	Enable additional Attributes as part of 29.510 Release 15.5	true/false	CNC Policy & PCF	Added in Release 1.6.x	
configmapApplicationConfig.profile.enableVirtualNrfResolution	enable virtual NRF session retry by Alternate routing service	true/false	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.virtualNrfFqdn	virtual NRF FQDN used to query static list of route	nrf.oracle.com	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.virtualNrfScheme	Scheme to be used with the virtual FQDN	http or https	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.virtualNrfPort	port number		CNC Policy & PCF	Added in Release 1.9.0	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
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 NRF-client.	integer value	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.nrfRetryConfig	Configurations required for the NRF Retry mechanism		CNC Policy & PCF	Added in Release 1.9.0	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
configmapApplicationConfig.profile.nrfRetryConfig.serviceRequestType	Specifies the type of service request.	• ALL_REQUESTS • AUTONOMOUS_NFREGISTER • AUTONOMOUS_NFSTATUS_SUBSCRIBE • AUTONOMOUS_NFUNSUBSCRIBE • AUTONOMOUS_NFSUBSCRIBE_UPDATE • AUTONOMOUS_NFDISCOVER • AUTONOMOUS_NFHEARTBEAT • AUTONOMOUS_NFPATCH • NFREGISTER • NFUPDATE • NF_STATUS_SUBSCRIBE • NFDISCOVER • NF_SUBSCRIBE_UPDATE • NF_UNSUBSCRIBE • NFDEREGISTER • NF_PROFILE_RETRIEVAL • NF_LIST_RETRIEVAL	CNC Policy & PCF	Added in Release 1.9.0	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
		Note: serviceRequestType : "ALL_REQUESTS" is the mandatory configuration and will be applicable to all serviceRequest types, but if custom config is required for any serviceRequestType then it can defined accordingly.			
configmapApplicationConfig.profile.nrfRetryConfig.primaryNRFRetryCount	Specifies the number of times a service request is retried to the primary NRF in case of failure.		CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.nrfRetryConfig.nonPrimaryNRFRetryCount	Specifies the number of times a service request is retried to the non-primary NRF in case of failure.		CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.nrfRetryConfig.alternateNRFRetryCount	Specifies the number of alternate NRFs that are retried in case of failure. When the value is specified as -1, all available NRF instances are tried.		CNC Policy & PCF	Added in Release 1.9.0	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
configmapApplicationConfig.profile.nrfRetryConfig.errorReasonsForFailure	Specifies the HTTP status codes or exceptions for which retry is attempted.	• All non 2xx HTTP status codes • SocketTimeoutException • JsonProcessingException • UnknownHostException • NoRouteToHostException	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.nrfRetryConfig.gatewayErrorCodes	Specifies the HTTP status codes sent by the Egress Gateway for which retry is attempted.	All HTTP Status codes	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.nrfRetryConfig.requestTimeout	Specifies the timeout period where no response is received from the Egress Gateway.	10 seconds	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.healthCheckConfig	Configurations required for the Health check of NRFs		CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.healthCheckConfig.healthCheckCount	Specifies the number of consecutive success or failures responses required to mark an NRF instance healthy or unhealthy.	-1,Values greater than 0. -1 denotes that the feature is disabled	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.healthCheckConfig.healthCheckInterval	Specifies the interval at which a health check of an NRF is performed.	5 seconds	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.healthCheckConfig.requestTimeout	Specifies the timeout period where no response is received from the Egress Gateway.	10 seconds	CNC Policy & PCF	Added in Release 1.9.0	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
configmapApplicationConfig.profile.healthCheckConfig.errorReasonsForFailure	Specifies the HTTP status codes or exceptions for which retry is attempted.	• 500 • 503 • 504 • SocketTimeoutException • JsonProcessingException • UnknownHostException • NoRouteToHostException	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.healthCheckConfig.gatewayErrorCodes	Specifies the HTTP status codes sent by the Egress Gateway for which retry is attempted.	All HTTP Status codes	CNC Policy & PCF	Added in Release 1.9.0	
configmapApplicationConfig.profile.supportedDataSetId	The data-set value to be used in queryParams for NFs autonomous/on-demand discovery.	POLICY	CNC Policy & PCF	Added in Release 1.7.1	
configmapApplicationConfig.profile.discoveryRefreshInterval	Defines the maximum ValidityPeriod for discovery results to be refreshed. The ValidityPeriod received in the discovery response shall be capped at this value. If ValidityPeriod received in discovery results is 60s, it will be capped to 10s as per configuration. If ValidityPeriod received in discovery results is 5s. No capping is applied and it is considered as 5s.	time in seconds	10	Added in Release 22.4.0	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
configmapApplicationConfig.profile.discoveryDurationBeforeExpiry	Defines the rate at which the NF shall resend discovery requests to NRF. If the discovery ValidityPeriod is 10s (after applying the capped value of discoveryRefreshInterval), then the discovery requests shall be sent at discoveryDurationBeforeExpiry * 10/100.	terms of percentage(1-100)	90	Added in Release 22.4.0	
configmapApplicationConfig.profile.enableDiscoveryRefresh	Flag to enable Automatic Discovery Refresh	true/false	false	Added in Release 22.4.0	
configmapApplicationConfig.profile.enableRediscoveryIfNoProdNFs	Flag to enable rediscovery when no producer NFs are available	true/false	false	Added in Release 22.4.0	
configmapApplicationConfig.profile.offStatesForRediscoveryIfNoProdNFs	Comma separated value for states to consider producer NFs as not available	SUSPENDED,UNDISCOVERABLE,DEREGISTERED	SUSPENDED,UNDISCOVERABLE,DEREGISTERED	Added in Release 22.4.0	
configmapApplicationConfig.profile.discoveryRetryInterval	Retry Interval after a failed autonomous discovery request	time	2000	Added in Release 22.4.0	
configmapApplicationConfig.profile.nrfRouteList	This attribute can be used when more than two NRFs are required to be configured. Either the primaryNrfApiRoot and secondaryNrfApiRoot OR this attribute can be used. If this attribute is to be used, useNrfRouteList can be set to true.		CNC Policy & PCF	Added in Release 23.1.0	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
configmapApplicationConfig.profile.useNrfRouteList	This attribute indicates that nrfRouteList can be used instead primaryNrfApiRoot and secondaryNrfApiRoot.	true/false	CNC Policy & PCF	Added in Release 23.1.0	

Here is a sample configuration for NRF client in `occnp_custom_values_24.3.0.yaml` file:

```
appinfo:
  serviceAccountName: ''
  # Set Infrastructure services to empty array if any one of below condition
is met
  #  1. Deploying on occne 1.4 or lesser version
  #  2. Not deploying on OCCNE
  #  3. Do not wish to monitor infra services such as db-monitor service
  # then the below mentioned attribute 'infra_services' should be uncommneted
and epmty array should be passed as already mentioned.
  #infraServices: []

perf-info:
  configmapPerformance:
    prometheus: ''


nrf-client:
  # This config map is for providing inputs to NRF-Client
  configmapApplicationConfig:
    # primaryNrfApiRoot - Primary NRF Hostname and Port
    # SecondaryNrfApiRoot - Secondary NRF Hostname and Port
    # retryAfterTime - Default downtime(in ISO 8601 duration format) of an
NRF detected to be unavailable.
    # nrfClientType - The NfType of the NF registering
    # nrfClientSubscribeTypes - the NFType for which the NF wants to
subscribe to the NRF.
    # appProfiles - The NfProfile of the NF to be registered with NRF.
    # enableF3 - Support for 29.510 Release 15.3
    # enableF5 - Support for 29.510 Release 15.5
    # renewalTimeBeforeExpiry - Time Period(seconds) before the Subscription
Validity time expires.
    # validityTime - The default validity time(days) for subscriptions.
    # enableSubscriptionAutoRenewal - Enable Renewal of Subscriptions
automatically.
    # acceptAdditionalAttributes - Enable additionalAttributes as part of
29.510 Release 15.5
    # enableVirtualNrfResolution=false
    # virtualNrfFqdn=nf1stub.ocpcf.svc:8080
    # virtualNrfScheme=http
    # virtualNrfPort=8080
```

```
    # requestTimeoutGracePeriod=2
    # nrfRetryConfig=[{ "serviceRequestType": "ALL_REQUESTS",
"primaryNRFRetryCount": 1, "nonPrimaryNRFRetryCount" : 1,
"alternateNRFRetryCount" : -1, "errorReasonsForFailure":
[503,504,500,"SocketTimeoutException","JsonProcessingException","UnknownHostEx
ception","NoRouteToHostException", "IOException"], "gatewayErrorCodes":
[503,429], "requestTimeout": 100 },{"serviceRequestType":
"AUTONOMOUS_NFREGISTER", "primaryNRFRetryCount": 1,
"nonPrimaryNRFRetryCount": 1, "alternateNRFRetryCount": -1,
"errorReasonsForFailure":
[503,504,500,"SocketTimeoutException","JsonProcessingException","UnknownHostEx
ception","NoRouteToHostException", "IOException"], "gatewayErrorCodes":
[503,429], "requestTimeout": 100 }]
    # healthCheckConfig={ "healthCheckCount": -1, "healthCheckInterval": 5,
"requestTimeout": 10, "errorReasonsForFailure":
[503,504,500,"SocketTimeoutException","JsonProcessingException","UnknownHostEx
ception","NoRouteToHostException", "IOException"], "gatewayErrorCodes":
[503,429] }
   profile: |-
    nrfRouteList=[{"nrfApi":"nrfDeployName-
nrf-1:8080","scheme":"http","weight":100,"priority":1},
{"nrfApi":"nrfDeployName-
nrf-2:8080","scheme":"http","weight":100,"priority":2},},
{"nrfApi":"nrfDeployName-
nrf-3:8080","scheme":"http","weight":100,"priority":3}]
    useNrfRouteList=true
    [appcfg]
    primaryNrfApiRoot=nrf1-api-gateway.svc:80
    secondaryNrfApiRoot=nrf2-api-gateway.svc:80
    nrfScheme=http
    retryAfterTime=PT120S
    nrfClientType=PCF
    nrfClientSubscribeTypes=CHF,UDR,BSF
    appProfiles=[{ "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b0123",
"nfSetIdList" = ["set1yz.pcfset.5gc.mnc012.mcc345",
"set1a.pcfset.5gc.mnc112.mcc345"] ,"nfType": "PCF", "nfStatus": "REGISTERED",
"plmnList": null, "nsiList": null, "fqdn": "occnp-ocpm-ingress-
gateway.ocpcf.svc", "interPlmnFqdn": null, "ipv4Addresses": null,
"ipv6Addresses": null, "priority": null, "capacity": null, "load": 80,
"locality": null, "pcfInfo": { "dnnList": [ "internet", "volte" ],
"supiRanges": [ { "start": "12123444444", "end": "232332323323232",
"pattern": null } ] }, "customInfo": null, "recoveryTime": null,
"nfServices": [ { "serviceInstanceId": "03063893-
cf9e-4f7a-9827-067f6fa9dd01", "serviceName": "npcf-am-policy-control",
"versions": [ { "apiVersionInUri": "v1", "apiFullVersion": "1.0.0", "expiry":
null } ], "scheme": "http", "nfServiceStatus": "REGISTERED", "fqdn": "occnp-
ocpm-ingress-gateway.ocpcf.svc", "interPlmnFqdn": null, "ipEndPoints": null,
"apiPrefix": null, "defaultNotificationSubscriptions": null, "allowedPlmns":
null, "allowedNfTypes": [ "AMF", "NEF" ], "allowedNfDomains": null,
"allowedNssais": null, "priority": null, "capacity": null, "load": null,
"recoveryTime": null, "supportedFeatures": null }, { "serviceInstanceId":
"03063893-cf9e-4f7a-9827-067f6fa9dd02", "serviceName": "npcf-
smpolicycontrol", "versions": [ { "apiVersionInUri": "v1", "apiFullVersion":
"1.0.0", "expiry": null } ], "scheme": "http", "nfServiceStatus":
"REGISTERED", "fqdn": "occnp-ocpm-ingress-gateway.ocpcf.svc",
"interPlmnFqdn": null, "ipEndPoints": null, "apiPrefix": null,
```

```
"defaultNotificationSubscriptions": null, "allowedPlmns": null,
"allowedNfTypes": [ "SMF", "NEF", "AF" ], "allowedNfDomains": null,
"allowedNssais": null, "priority": null, "capacity": null, "load": null,
"recoveryTime": null, "supportedFeatures": null }, { "serviceInstanceId":
"03063893-cf9e-4f7a-9827-067f6fa9dd03", "serviceName": "npcf-ue-policy-
control", "versions": [ { "apiVersionInUri": "v1", "apiFullVersion": "1.0.0",
"expiry": null } ], "scheme": "http", "nfServiceStatus": "REGISTERED",
"fqdn": "occnp-ocpm-ingress-gateway.ocpcf.svc", "interPlmnFqdn": null,
"ipEndPoints": null, "apiPrefix": null, "defaultNotificationSubscriptions":
null, "allowedPlmns": null, "allowedNfTypes": [ "AMF" ], "allowedNfDomains":
null, "allowedNssais": null, "priority": null, "capacity": null, "load":
null, "recoveryTime": null, "supportedFeatures": null } ]}]
    enableF3=true
    enableF5=true
    renewalTimeBeforeExpiry=3600
    validityTime=30
    enableSubscriptionAutoRenewal=true
    nfHeartbeatRate=80
    acceptAdditionalAttributes=false
    supportedDataSetId=POLICY
    discoveryRefreshInterval=10
    discoveryDurationBeforeExpiry=90
    enableDiscoveryRefresh=false
    enableRediscoveryIfNoProdNFs=false
    offStatesForRediscoveryIfNoProdNFs=SUSPENDED,UNDISCOVERABLE,DEREGISTERED
    discoveryRetryInterval=2000
```

> **Note**
>
> For using TLS during deployment, the value of the `profile.nrfScheme` and `appProfiles.scheme` parameters must be set to https.

Table 3-47 Configurable Parameters for nrf-client-nfdiscovery

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
nrf-client.nrf-client-nfdiscovery.configmapApplicationConfig	This config map is used to provide inputs to NRF Client for NF discovery.	Yes	Not Applicable	CNC Policy & PCF	Added in Release 1.14.0
nrf-client.nrf-client-nfdiscovery.readinessProbe.httpGet.port	Specifies the port to be used for readiness probes.	Yes	9000	CNC Policy & PCF	Added in 24.10
nrf-client.nrf-client-nfdiscovery.livenessProbe.httpGet.port	Specifies the port to be used for liveness probes.	Yes	9000	CNC Policy & PCF	Added in 24.10

Here is a sample configuration for NRF client in `occnp_custom_values_24.3.0.yaml`

```
nrf-client:
```

```
nrf-client-nfdiscovery:
 readinessProbe:
  httpGet:
   port: *containerMonitoringHttp
 livenessProbe:
  httpGet:
   port: *containerMonitoringHttp
```

Table 3-48 Configurable Parameters for nrf-client-nfmanagement

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
nrf-client.nrf-client-nfmanagement.configmapApplicationConfig	This config map is used to provide inputs to NRF Client for NF management.	Yes	Not Applicable	CNC Policy & PCF	Added in Release 1.14.0
nrf-client.nrf-client-nfmanagement.readinessProbe.httpGet.port	Specifies the port to be used for readiness probes.	Yes	9000	CNC Policy & PCF	Added in 24.1.0
nrf-client.nrf-client-nfmanagement.livenessProbe.httpGet.port	Specifies the port to be used for liveness probes.	Yes	9000	CNC Policy & PCF	Added in 24.1.0

Here is a sample configuration for NRF client in `occnp_custom_values_24.3.0.yaml`

```
nrf-client:
 nrf-client-nfmanagement:
  readinessProbe:
   httpGet:
    port: *containerMonitoringHttp
  livenessProbe:
   httpGet:
    port: *containerMonitoringHttp
```

3.12 PCRF-Core Configurations

This section describes the customizatons that is made in `occnp_custom_values_24.3.0.yaml` file to customize Pcrf-core configurations.

Table 3-49 Configurable Parameters for Pcrf-core Configuration

Parameter	Description	Mandatory/ Optional Paramete	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
pcrf-core.envMysqlDatabase	Database name the pcrf-core service will connect to.	Yes	Not applicable	CNC Policy, PCF, & PCRF	Added in Release 1.7.1	--

Table 3-49 (Cont.) Configurable Parameters for Pcrf-core Configuration

Parameter	Description	Mandatory/ Optional Paramete	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
pcrf-core.envDiameterRealm	Diameter Realm of PCRF	Yes	Not applicable	CNC Policy, PCF, & PCRF	Added in Release 1.7.1	Applicable only when diameter gateway is enabled. **Note:** Example: oracle.com
pcrf-core.envDiameterIdentity	Diameter Host of PCRF diameter gateway	Yes	Not applicable	CNC Policy, PCF, & PCRF	Added in Release 1.7.1	Applicable only when diameter gateway is enabled. **Note:** Example: oc-diam-gateway
pcrf-core.envDbQueryTimeout	Database Query Timeout	Yes	0	CNC Policy, PCF, & PCRF	Added in Release 22.4.5	Represents a JDBC statement timeout, in milliseconds. When timeouts are set, the driver would wait for the given number of seconds for the query to execute (i.e. executeQuery and executeUpdate) and throw an SQLTimeout Exception if there is no response within that time. **Note:** It is recommended to set this value to zero during install/ upgrade.

Table 3-49 (Cont.) Configurable Parameters for Pcrf-core Configuration

Parameter	Description	Mandatory/ Optional Paramete	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
enableLateArrival	Indicates whether the stale request cleanup functionality for PCRF Core is enabled or disabled.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 24.3.0	--
enableLateProcessing	Indicates whether the stale request cleanup functionality for PCRF Core is enabled or disabled.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 24.3.0	--
sbiTimerEnabled	Enables PCRF-Core handling of HTTP SBI headers at incoming and outgoing requests. These SBI headers are `3gpp-Sbi-Origination-Timestamp`, `3gpp-Sbi-Max-Rsp-Time`, and `3gpp-Sbi-Sender-Timestamp`.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 24.3.0	--
skipLateProcessingForTerminate	When the value of this parameter is set to `true` , PCRF-Core will not check if Terminate requests have gone stale.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 24.3.0	--

Here is a sample configuration in `occnp_custom_values_24.3.0.yaml` file:

```
pcrf-core: # database name core service will connect to
  envMysqlDatabase: occnp_pcrf_core
  envDiameterRealm: ''
  envDiameterIdentity: 'pcrf-core'
  envDbQueryTimeout: 2000
```

Load Shedding through Admission Control in PCRF-Core

Important

These advanced configurations must not be used without consulting My Oracle Support (https://support.oracle.com).

Table 3-50 Advanced Configuration for Load Shedding

Name	Description	Type	Notes/Examples
ADMISSION.Level <i>.BusyThreshold	The number of outstanding messages required to enter this level of busy.	Int	Key: ADMISSION.Level***1***. BusyThresholdValue: 300 **Note:** “i” represents the busy level number.
ADMISSION.Level <i>.BusyTime	The minimum amount of time (in milliseconds) the system needs to have crossed the busy threshold before entering this level of busy.	Int	Key: ADMISSION.Level***1***. BusyTimeValue: 300 **Note:** “i” represents the busy level number.
ADMISSION.Level <i>.ClearThreshold	The maximum number of outstanding messages allowed to clear this level of busy.	Int	Key: ADMISSION.Level***1***. ClearThresholdValue: 150 **Note:** “i” represents the busy level number.
ADMISSION.Level <i>.ClearTime	The minimum amount of time (in milliseconds) the system needs to have crossed the clear threshold before clearing this level of busy.	Int	Key: ADMISSION.Level***1***. ClearTimeValue: 500 **Note:** “i” represents the busy level number.

Table 3-50 (Cont.) Advanced Configuration for Load Shedding

Name	Description	Type	Notes/Examples
ADMISSION.Level <i>.Action	Action to apply to any messages not matching any filters at this busy level. The possible values for Action are: • DROP • Name of a Result-Code or Experimental-Result-Code (e.g. DIAMETER_TOO_BUSY) Custom Result-Code or Experimental-Result-Code entered as vendorid:code (e.g. 10415:5011).	Int	Key: ADMISSION.Level***1***.ActionValue: DIAMETER_TOO_BUSY **Note:** “i” represents the busy level number.
ADMISSION.Level <i>.DiameterRule<j>.Filter	Filter to apply when determining which messages match this rule and should have the defined action applied. ”j” represents the rule number. “j” shall start at 1 for the first rule and increment monotonically by 1 for each subsequent rule. The syntax of the filter is as follows: <AppName>[/<MsgName>[/<AVPList>]]. The brackets denote “optionality”. As such, the MsgName and AVPListare optional. The “/” (slash) is used as a delimiter. “AppName” is the name of the application (e.g. Gx) MsgName is the name of the message (e.g. CCR) “AVPList” has the following syntax: *[<AVPName><Operand><AVPValue> [&&]]. “AVPName” is the name of the AVP (e.g. Called-Station-Id). “Operand” has two possible values: “=” or “!=”. “AVPValue” is the value of the AVP. An example of AVPList is: “CC-Request-Type=1 && Called-Station-Id=IMS” An example of a filter is: “Gx/CCR/CC-Request-Type=1 && Called-Station-Id=IMS”	Int	Key: ADMISSION.Level***1***.DiameterRule***1***.Filter Value: Gx/CCR/CC-Request-Type=1 Key: ADMISSION.Level***1***.DiameterRule***2***.Filter Value: Gx/CCR/CC-Request-Type=1 && Called-Station-Id=ims Key: ADMISSION.Level***2***.DiameterRule***1***.Filter Value: Rx/AAR/Rx-Request-Type=0 **Note:** “i” represents the busy level number.

Table 3-50 (Cont.) Advanced Configuration for Load Shedding

Name	Description	Type	Notes/Examples
ADMISSION.Level<i>.DiameterRule<j>.Action	Action to apply to any messages matching the rule's filter when the system is in this level of busy. The possible values for Action are: · DROP · Name of a Result-Code or Experimental-Result-Code (e.g. DIAMETER_TOO_BUSY) · Custom Result-Code or Experimental-Result-Code entered as vendorid:code (e.g. 10415:5011).	--	Key: ADMISSION.Level***1***.DiameterRule***1***.Action Value: DIAMETER_TOO_BUSY Key: ADMISSION.Level***1***.DiameterRule***2***.Action Value: DROP Key: ADMISSION.Level***2***.DiameterRule***1***.Action Value: ACCEPT **Note:** "i" represents the busy level number.

3.13 Binding Service Configurations

This section describes the customizatons that is made in `occnp_custom_values_24.3.0.yaml` file to customize Binding service configurations.

Table 3-51 Configurable Parameters for Binding service Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
binding.envMysqlDatabase	Database name the binding service will connect to	Mandatory	Not applicable	CNC Policy, PCF, & PCRF	Added in Release 1.7.1
binding.sessionDeleteDelaySeconds	Binding sends deleteSession to PCF after delay (default 3s), hence, retaining session until needed.	Optional	3 (seconds)	CNC Policy, PCF, & PCRF	Added in Release 22.2.2

Here is a sample configuration in `occnp_custom_values_24.3.0.yaml` file:

```
binding:
  envMysqlDatabase: occnp_binding
```

```
envMysqlDatabaseConfigServer: *configServerDB
bsfEnabled: false
replicas: 2
bindingThreadCap: 120
bindingQueueCapacity: 128
http2ServerMaxConcurrentStream: 100
mySql:
  connection:
    minIdle: 120
    maxPoolSize: 120
sessionDeleteDelaySeconds:3
```

3.14 Audit Service Configuration

This section describes the customizatons that you should make in custom-value.yaml file to customize Audit service configurations.

Table 3-52 Configurable Parameters for Audit Service Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
sm-service.auditSmSessionTtl	SM Policy Association normal age	No	86400	CNC Policy & PCF	Added in Release 1.6.x	Specifies age of a SM policy association after which a record is considered to be stale on PCF and the SMF is queried for presence of such associations. Applicable only when SM service is enabled.

Table 3-52 (Cont.) Configurable Parameters for Audit Service Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
sm-service.auditSmSessionMaxTtl	SM Policy Association maximum age	No	172800	CNC Policy & PCF	Added in Release 1.6.x	Specifies maximum age of a SM Policy Association after which a record is purged from PCF SM database without sending further queries to SMF. Applicable only when SM service is enabled.

Here is a sample configuration in custom-values.yaml.file:

```
sm-service:
  auditSmSessionTtl: 86400
  auditSmSessionMaxTtl: 172800
```

3.15 Diameter Gateway and Diameter Connector Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` file to customize Diameter configurations.

Table 3-53 Configurable Parameters for Diameter Gateway Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
diam-gateway.envMysqlDatabaseConfigServer	Specifies the name of the database for Config server service.	Yes	occnp_config_server	CNC Policy, PCF, & PCRF	Added in 1.14.0	
diam-gateway.envDiameterRealm	Diameter Realm of PCF diameter gateway	Yes	Not applicable	CNC Policy, PCF, & PCRF	Added in Release 1.7.1	example: oracle.com Applicable only when diameter gateway is enabled.
diam-gateway.envDiameterIdentity	Diameter Host of PCF diameter gateway	Yes	Not applicable	CNC Policy, PCF, & PCRF	Added in Release 1.7.1	example: oc-diam-gateway Applicable only when diameter gateway is enabled.
diam-gateway.envDiameterHostIp	Contains all the k8s cluster worker node names and corresponding IP addresses in the following format: NodeName1=<ip1>,NodeName2=<ip2> If LoadBalancer is being used, provide the LoadBalancer IP.	Optional		CNC Policy, PCF, & PCRF	Added in Release 1.12.0	
diam-gateway.envDbConnStatusHttpEnabled	To monitor the database service connectivity status, set the value for this parameter to true.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 1.14.0	

Table 3-53 (Cont.) Configurable Parameters for Diameter Gateway Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
diam-gateway.envSupportedIpAddressType	This parameter specifies the IP address type to be configured as diameter peer nodes. When the value is specified as IPv4, hosts with IPv4 address type are configured as diameter peer nodes and hosts with IPv6 address type are ignored. When the value is specified as IPv6, hosts with IPv6 address type are configured as diameter peer nodes and hosts with IPv4 address type are ignored. To configure hosts with both IPv4 and IPv6 address types, set the value for this parameter as `Both`.	Mandatory	IPv4	CNC Policy, PCF, & PCRF	Added in Release 1.14.1	The values are not case-sensitive. Supported values are: • IPV4 • IPV6
diam-gateway.envDiameterValidationStrictParsing	This parameter enables or disables the strict parsing.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 23.2.0	NA
ENABLE_LATE_ARRIVAL	Indicates whether the stale request cleanup functionality for PCRF Core is enabled or disabled.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 24.3.0	
ENABLE_LATE_PROCESSING	Indicates whether the stale request cleanup functionality for PCRF Core is enabled or disabled.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 24.3.0	

Table 3-53 (Cont.) Configurable Parameters for Diameter Gateway Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
LATE_ARRIVAL_MAX_RESPONSE_TIME	Defines the default duration (in milliseconds) for which Diameter Gateway waits for a response in case the Diameter request does not contain the required AVP.	Optional		CNC Policy, PCF, & PCRF	Added in Release 24.3.0	
SKIP_LATE_PROCESSING_FOR_TERMINATE	When enabled, PCRF-Core will not check if Terminate requests have gone stale.	Optional	false	CNC Policy, PCF, & PCRF	Added in Release 24.3.0	

Here is a sample configuration in `occnp_custom_values_24.3.0.yaml` file:

```
diam-gateway:
  envMysqlDatabaseConfigServer: *configServerDB
  envDiameterRealm: 'oracle.com'
  envDiameterIdentity: 'oc-diam-gateway'
  #This should contain all the k8s cluster worker node name and ip
corresponding to it in a format i.e. NodeName1=<ip1>,NodeName2=<ip2>
  #If LoadBalancer is being used then give all ip as LoadBalancer's ip
  envDiameterHostIp: ''
  envDbConnStatusHttpEnabled: false
  envSupportedIpAddressType: 'IPv4'

  staticIpAddress: ''
  staticDiamNodePort: *svcDiamGatewayDiamNodePort

  deployment:
    customExtension:
      annotations: {
          # Enable this section for service-mesh based installation
  #          traffic.sidecar.istio.io/excludeOutboundPorts: "9000,5801",
  #          traffic.sidecar.istio.io/excludeInboundPorts: "9000,5801"
      }
```

The `lbService` provides the annotations and labels for service diameter gateway and the `nonlbService` provides annotations and labels for headless diameter gateway.

Table 3-54 Configurable Parameters for Diameter Connector Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
diam-connector.envDiameterRealm	Diameter Realm of PCF	Yes	Not applicable	CNC Policy & PCF	Added in Release 1.6.x	example: oracle.com Applicable only when diameter connector is enabled.
diam-connector.envDiameterIdentity	Diameter Host of PCF	Yes	Not applicable	CNC Policy & PCF	Added in Release 1.6.x	example: ocpcfApplicable only when diameter connector is enabled.
diam-connector.envMysqlDatabaseConfigServer	Specifies the name of the database for Config server service.	Yes	occnp_config_server	CNC Policy, PCF, & PCRF	Added in Release 1.15.0	
diam-connector.envSyEnableSubsIdOnSTR	Determines whether to include Subscription-Id information in Subscription-Id AVPs when sending a STR Message.	Mandatory	false	CNC Policy, PCF, & PCRF	Added in Release 23.2.0	
diam-connector.timer.reconnectDelay	Specifies the time frame to delay before attempting to reconnect after a connection failure in seconds.	M	3	CNC Policy, PCF, & PCRF	Added in Release 24.1.0	
diam-connector.timer.responseTimeout	Specifies the response timeout interval in seconds.	M	4	CNC Policy, PCF, & PCRF	Added in Release 24.1.0	
diam-connector.timer.connectionTimeOut	Specifies the connection timeout interval in seconds.	M	3	CNC Policy, PCF, & PCRF	Added in Release 24.1.0	
diam-connector.timer.watchdogInterval	Specifies the watchdog interval in seconds.	M	6 seconds	CNC Policy, PCF, & PCRF	Added in Release 24.1.0	

Table 3-54 (Cont.) Configurable Parameters for Diameter Connector Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
diam-connector.defaultBoundedElasticSize	Specifies the default maximum size for the boundedElastic() scheduler.	O	120 **Note**: When the configured value is less than or equal to 0, then default values from reactor library are applied.	CNC Policy, PCF, & PCRF	Added in Release 24.2.1	
diam-connector.defaultBoundedElasticQueueSize	Specifies the default maximum number of enqueued tasks per thread for the global boundedElastic() scheduler, initialized by system property reactor.	O	64 **Note**: When the configured value is less than or equal to 0, then default values from reactor library are applied.	CNC Policy, PCF, & PCRF	Added in Release 24.2.1	

Here is a sample configuration in `occnp_custom_values_24.3.0.yaml` file:

```
diam-connector:
  envDiameterRealm: 'oracle.com'
  envDiameterIdentity: 'ocpcf'
  envMysqlDatabaseConfigServer: *configServerDB
  envSyEnableSubsIdOnSTR: false
  # This timer configuration is used in configMap of diam-conn
  timer:
    reconnectDelay: 3
    responseTimeout: 4
    connectionTimeOut: 3
    watchdogInterval: 6
```

3.16 BSF Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` file to customize default BSF configurations.

Table 3-55 Configurable Parameters for BSF Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
sm-service.defaultBsfApiRoot	Api root of pre-configured BSF	No	Not applicable	CNC Policy & PCF	Added in Release 1.5.x	Applicable only when SM service is enabled. Required, if PCF uses pre-configured BSF. For Example: "https://bsf.apigateway:8001/"
binding.bsfEnabled	Enable/Disable the binding operation (register and deregister) with the BSF	No	False	CNC Policy & PCF	Added in Release 1.7.1	Applicable only when Binding service is enabled.

Here is a sample configuration in `occnp_custom_values_24.3.0.yaml` file:

```
sm-service:
  defaultBsfApiRoot: 'https://bsf.apigateway:8001'


binding:
    bsfEnabled: false
```

3.17 Kubernetes Service Account Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` file to customize kubernetes service account configurations.

Table 3-56 Configurable Parameters for Kubernetes Service Account Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ldap-gateway.serviceAccountName	K8s Service Account to access (RBAC) the K8s API server to retrieve status of PCF services and pods. The account should have read access ("get" , "watch" , "list") to pods, services and nodes.	Conditional	Not applicable	CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	

Here is a sample configuration in `occnp_custom_values_24.3.0.yaml` file:

```
ldap-gateway:
  serviceAccountName: ''
```

3.18 API Root Configuration for Resource URI and Notification URI

This section describes the configuration parameters that can be used to API Root configuration.

To configure these parameters, you should configure the following configurable parameters in the `occnp_custom_values_24.3.0.yaml` file:

Table 3-57 Configurable Parameters for Api Root Configuration for Notification URI

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.pcfApiRoot	API root of PCF that is used in • Notification URI generated by PCF when sending request to other producer NFs (like NRF, UDR, CHF, etc..) • Resource URI generated by PCF, on successful creation of policy association for requests from SMF, AMF, and UE.	No	Ingress gateway service name and port	CNC Policy & PCF	Added in Release 1.5.x	If not configured then the ingress gateway service name and port will be used as default value. In case of `cnlb` is not enabled then configure pcfApiRoot as : `https://<Helm namespace>-pcf-ingress-gateway:443` If `cnlb` is enabled, then ensure pcfApiRoot has to be set with value of `http://"<cnlbIp>":"<cnlbPort>` to enable ingress routing.

Table 3-57 (Cont.) Configurable Parameters for Api Root Configuration for Notification URI

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.deploymentNrfClientService.nfApiRoot	API root of PCF	Mandatory	Not Applicable	CNC Policy & PCF	Added in Release 1.6.x	Applicable only when NRF Client services are enabled. Value of this parameter should be same as the value of "global.pcfApiRoot" parameter. However, if the user has not configured `pcfApiRoot`, it is required to provide the values for Ingress Gateway service name and port manually. **Example**: https://<Helm namesp

Table 3-57 (Cont.) Configurable Parameters for Api Root Configuration for Notification URI

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
						ace>-pcf-ingress-gateway:80

3.19 Basic Configurations in Ingress Gateway

This section describes the configuration parameters that are required for basic configurations in Ingress Gateway.

> **Note**
>
> Following configurations are applicable only when ingress-gateway is enabled.

Table 3-58 Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.metalLbIpAllocationEnabled	Enable or disable IP Address allocation from Metallb Pool	No	false	CNC Policy, PCF, cnPCRF	Added in Release 1.5.x
global.metalLbIpAllocationAnnotation	Address Pool Annotation for Metallb	No	"metallb.universe.tf/address-pool: signaling"	CNC Policy, PCF, cnPCRF	Added in Release 1.5.x
ingress-gateway.enableIncomingHttp	Enable it to accept incoming http requests	No	False	CNC Policy, PCF, cnPCRF	Added in Release 1.5.x
ingress-gateway.ingressServer.keepAlive.enabled		No	false		Added in Release 1.7.3
ingress-gateway.ingressServer.keepAlive.idealTime		No	180 (in seconds)		Added in Release 1.7.3

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.ingressServer.keepAlive.count		No	9		Added in Release 1.7.3
ingress-gateway.ingressServer.keepAlive.interval		No	60 (in seconds)		Added in Release 1.7.3
ingress-gateway.isIpv6Enabled	Set the value to true for this parameter when NF is deployed in IPv6 cluster.	No	false		Added in Release 1.14.0
global.staticIpAddressEnabled	set to value to true to enable it	No	false	Converged Policy and PCF	Added in Release 23.2.0
global.staticIpAddress	set static load balancer IP, else a random IP will be assigned by the External LoadBalancer from its IP Pool.	No	NA	Converged Policy and PCF	Added in Release 23.2.0
ingress-gateway.applicationThreadPoolConfig.corePoolSize	It is preferred to use fixed size thread pool as this ensures all threads are created during startup as thread creation during runtime is expensive and can have impact on performance. This parameter indicates the minimum number of workers to keep alive without timing out. For details on the recommended application thread pool configuration, see Table 3-59.	No	8		Added in Release 23.3.0

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.applicationThreadPoolConfig.maxPoolSize	This defines the maximum number of threads that can ever be created. To create fixed size thread pool, corePoolSize and maxPoolSize should be same. For details on the recommended application thread pool configuration, see Table 3-59.	No	8		Added in Release 23.3.0
ingress-gateway.applicationThreadPoolConfig.queueCapacity	This indicates the number of tasks in the queue when all core pools are filled. Threads will be scalable to maximum pool size when queue is full. For details on the recommended application thread pool configuration, see Table 3-59.	No	1000		Added in Release 23.3.0

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
global.TRACE_ID_GENERATION_ENABLED	Used to enable or disable the addition of the unique log identifier (ocLogId) in the log messages. By default, the value of this parameter is set to true, and the feature is enabled. When this feature is enabled, Ingress Gateway generates the ocLogId and propagates the headers to the succeeding microservices. Ingress Gateway passes the ocLogId generated as the header to all the backend services such as SM service, AM service, UE Policy service, UDR Connector, CHF Connector, and SOAP Connector.	Yes	true	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.enableIncomingHttps	To enable HTTPS for ingress traffic.	Mandatory	false	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.privateKey.k8SecretName	Name of the Kubernetes Secret which contains the private key for Policy.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.privateKey.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the private key for Policy can be found	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.privateKey.rsa.fileName	File name for Policy's private key generated using the RSA algorithm	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.service.ssl.certificate.k8SecretName	Name of the Kubernetes Secret which contains the Policy Certificate.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.certificate.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the Policy Certificate can be found.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.certificate.rsa.fileName	File name for Policy's Certificate, generated using an RSA resources.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.caBundle.k8SecretName	Name of the Kubernetes Secret which contains the Trust Chain Certificate.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.caBundle.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the Trust Chain Certificate can be found.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.caBundle.fileName	File name for the Trust Chain Certificate	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.keyStorePassword.k8SecretName	Name of the Kubernetes Secret which contains the Key Store Password file	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.keyStorePassword.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the Key Store Password file can be found.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.keyStorePassword.fileName	File name that has password for keyStore	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.service.ssl.trustStorePassword.k8SecretName	Name of the Kubernetes Secret which contains the Trust Store Password file.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.trustStorePassword.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the Trust Store Password file can be found.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.trustStorePassword.fileName	File name that has password for trustStore.	Mandatory	Not Applicable	CNC Policy, PCF, cnPCRF	24.1.0
ingress-gateway.service.ssl.tlsVersion	Indicates the TLS version.	Mandatory	**Data Type**: String **Default Value**: TLSv1.2 **Range**: • TLSv 1.2 • TLSv 1.3	CNC Policy, PCF, cnPCRF	24.1.0

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.allowedCipherSuites	Indicates the allowed Ciphers suites.	Optional	**Data Type**: String **Default Value**: NA **Range**: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECD	CNC Policy, PCF, cnPCRF	24.1.0

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
			HE_RSA_WITH_AES_128_GCM_SHA256		

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.cipherSuites	Indicates the supported cipher suites.	Optional	**Data Type**: String **Default Value**: NA **Range**: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECD	CNC Policy, PCF, cnPCRF	24.1.0

Table 3-58 (Cont.) Configurable Parameters for Basic Configurations in Ingress Gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
			HE_RSA_WITH_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • TLS_CHACHA20_POLY1305_SHA256		

Here is a sample configuration for configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

```
ingress-gateway:
 #Cipher Suites to be enabled on client side
  cipherSuites:
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

  allowedCipherSuites:
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
```

```
  #keep alive settings
  ingressServer:
    keepAlive:
      enabled: false
      idealTime: 180  #in seconds
      count: 9
      interval: 60 #in seconds

  #Enabled when deployed in Ipv6 cluster
  isIpv6Enabled: false

  #It is preferred to use fixed size thread pool as this ensures all threads
are
  #created during startup as thread creation during runtime is expensive and
can
  #have impact on performance. To create fixed size thread pool, corePoolSize
&
  #maxPoolSize should be same.
  applicationThreadPoolConfig:
      corePoolSize: 8
      maxPoolSize: 8
      queueCapacity: 1000

  # Enable it to accept incoming http requests
  enableIncomingHttp: true

  # ---- HTTPS Configuration - BEGIN ----
  enableIncomingHttps: false

  service:
    ssl:
      tlsVersion: TLSv1.2
      #supportedCipherSuiteList: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
      privateKey:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        rsa:
          fileName: rsa_private_key_pkcs1.pem
      certificate:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        rsa:
          fileName: ocegress.cer
      caBundle:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        fileName: caroot.cer
      keyStorePassword:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        fileName: key.txt
      trustStorePassword:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        fileName: trust.txt
```

```
# Enable or disable IP Address allocation from Metallb Pool
metalLbIpAllocationEnabled: false

# Address Pool Annotation for Metallb
metalLbIpAllocationAnnotation: "metallb.universe.tf/address-pool: signaling"
# -----Ingress Gateway Settings - END-----
```

Table 3-59 Recommended Application Threadpool Configuration

Traffic towards 1 Pod (TPS)	corePoolSize	maxPoolSize	queueCapacity
500	8	8	1000
1000	8	8	1800
1500	16	16	2500
2000	16	16	3300

3.20 Basic Configurations in Egress Gateway

This section describes the configuration parameters that are required for basic configurations in Egress Gateway.

> **Note**
>
> Following configurations are applicable only when Egress-gateway is enabled.

Table 3-60 Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.enableForwardedHeader	Enabling this parameter, egress-gateway will add Forwarded and x-Forwaredheaders	Optional	false	CNC Policy & PCF	Added in Release 1.8.3
egress-gateway.isIpv6Enabled	Set the value to true for this parameter when NF is deployed in IPv6 cluster.	Optional	false	CNC Policy & PCF	Added in Release 1.14.0
egress-gateway.http1.enableOutgoingHTTP1	Set the value for this parameter to true to enable Egress HTTP1.1 requests.	Optional	false	CNC Policy & PCF	Added in Release 22.2.0
egress-gateway.userAgentHeaderConfigMode	This parameter is used to govern the user-agent configurations from Helm or REST.	Optional	HELM	CNC Policy & PCF	

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.userAgentHeader.enabled	Specifies whether the feature is enabled or disabled.	Optional	false	CNC Policy & PCF	
egress-gateway.userAgentHeader.nfType	This parameter holds the nfType that will be used to generate the user agent header.	Optional	PCF	CNC Policy & PCF	
egress-gateway.userAgentHeader.nfInstanceId	This parameter represents the UUID of the CNPCF deployment that will be used to generate the user agent header.	Optional	empty string	CNC Policy & PCF	
egress-gateway.userAgentHeader.addFqdnToHeader	This parameter specifies if the user agent will use the FQDN information under the module to append it when generating the user agent header.The default value is set to 'false' meaning that the FQDN information will not be encoded into the user agent header during its generation.	Optional	false	CNC Policy & PCF	
egress-gateway.userAgentHeader.nfFqdn	This is an optional parameter and can be present or not, if operators want to include the FQDN string configured under this section then the parameter userAgentHeader.addFqdnToHeader needs to be enabled.	Optional	empty string	CNC Policy & PCF	

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.userAgentHeader.overwriteHeader	This parameter is used to govern if we want to include the User-Agent header generated at CNPCF Egress Gateway or forward the User-Agent received from service request. By default it will be set to true as CNPCF always generates its own service requests.	Optional	true	CNC Policy & PCF	
egress-gateway.sniHeader.enabled	By enabling this parameter, egress-gateway will add SNI flag in client hello message of outbound traffic. **Note:** SNI enabling is depending on the initssl parameter from egress-gateway helm charts (Default value of initssl=true[TLS enable] , initssl=false[TLS disable]) .	Optional	false	CNC Policy & PCF	Added in release 23.2.0
egress-gateway.enableOutgoingHttps	This parameter is used to enable HTTPS for egress traffic.	Mandatory	false	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.egressGwCertReloadEnabled	This parameter is used to enable reloading the gateway certificate.	Mandatory	false	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.egressGwCertReloadPath	Accepts a valid reload path.	Mandatory	/egress-gw/store/reload	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.privateKey.k8SecretName	Name of the Kubernetes Secret which contains the private key for PCF.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.service.ssl.privateKey.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the private key for PCF can be found	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.privateKey.rsa.fileName	File name for PCF's private key generated using the RSA algorithm	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.privateKey.ecdsa.fileName	File name for PCF's private key generated using the ECDSA algorithm	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.certificate.k8SecretName	Name of the Kubernetes Secret which contains the PCF Certificate.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.certificate.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the PCF Certificate can be found.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.certificate.rsa.fileName	File name for PCF's Certificate generated using an RSA resources.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.certificate.ecdsa.fileName	File name for PCF's Certificate, generated using an ECDSA resources.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.caBundle.k8SecretName	Name of the Kubernetes Secret which contains the Trust Chain Certificate.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.caBundle.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the Trust Chain Certificate can be found.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.service.ssl.caBundle.fileName	File name for the Trust Chain Certificate	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.keyStorePassword.k8SecretName	Name of the Kubernetes Secret which contains the Key Store Password file.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.keyStorePassword.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the Key Store Password file can be found.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.keyStorePassword.fileName	File name that has password for keyStore	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.trustStorePassword.k8SecretName	Name of the Kubernetes Secret which contains the Trust Store Password file.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.trustStorePassword.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the Trust Store Password file can be found.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.trustStorePassword.fileName	File name that has password for trustStore.	Mandatory	Not applicable	CNC Policy & PCF	Added in release 24.1.0
egress-gateway.service.ssl.tlsVersion	Indicates the TLS version.	Mandatory	**Data Type**: String **Default Value**: TLSv1.2 **Range**: • TLSv 1.2 • TLSv 1.3	CNC Policy, PCF, &cnPCRF	Added in release 24.1.0

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.allowedCipherSuites	Indicates the allowed Ciphers suites.	Optional	**Data Type**: String **Default Value**: NA **Range**: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_	CNC Policy, PCF, &cnPCRF	Added in release 24.1.0

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
			WITH_AES_128_GCM_SHA256		

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.cipherSuites	Indicates the supported cipher suites.	Optional	**Data Type**: String **Default Value**: NA **Range**: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_	CNC Policy, PCF, &cnPCRF	Added in release 24.1.0

Table 3-60 (Cont.) Configurable Parameters for Basic Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
			WITH_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • TLS_CHACHA20_POLY1305_SHA256		

Here is a sample configuration for configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

```
egress-gateway:
  # enabling this egress-gateway will add Forwarded and x-Forwaredheaders
  enableForwardedHeader: false

  #Enabled when deployed in Ipv6 cluster
  isIpv6Enabled: false

  #---- User-Agent header generation configuration - BEGIN ----
  userAgentHeaderConfigMode: HELM
  userAgentHeader:
    enabled: false # flag to enable or disable the feature
    nfType: "PCF" # NF type of consumer NF
    nfInstanceId: "" # NF type of consumer NF
    addFqdnToHeader: false # Flag to add fqdn. If enabled then user-agent
header will be generated along with the fqdn configured otherwise fqdn will
not be added
    nfFqdn: "" #fqdn of NF. This is not the fqdn of gateway
    overwriteHeader: true
  #---- User-Agent header generation configuration - END ----
```

```
#Cipher Suites to be enabled on client side
cipherSuites:
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

allowedCipherSuites:
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

# ---- HTTPS Configuration - BEGIN ----

#Enabling it for egress https requests
enableOutgoingHttps: false

egressGwCertReloadEnabled: false
egressGwCertReloadPath: /egress-gw/store/reload

service:
  ssl:
    tlsVersion: TLSv1.2
    privateKey:
      k8SecretName: ocpcf-gateway-secret
      k8NameSpace: ocpcf
      rsa:
        fileName: rsa_private_key_pkcs1.pem
      ecdsa:
        fileName: ssl_ecdsa_private_key.pem
    certificate:
      k8SecretName: ocpcf-gateway-secret
      k8NameSpace: ocpcf
      rsa:
        fileName: ocegress.cer
      ecdsa:
        fileName: ssl_ecdsa_certificate.crt
    caBundle:
      k8SecretName: ocpcf-gateway-secret
      k8NameSpace: ocpcf
      fileName: caroot.cer
    keyStorePassword:
      k8SecretName: ocpcf-gateway-secret
      k8NameSpace: ocpcf
      fileName: key.txt
    trustStorePassword:
      k8SecretName: ocpcf-gateway-secret
      k8NameSpace: ocpcf
      fileName: trust.txt
# ---- HTTPS Configuration - END ----

#Enabling it for egress http1.1 requests
```

```
http1:
  enableOutgoingHTTP1: false # Flag to enable or disable the feature
```

3.21 Service and Container Port Configuration

This section describes the customizatons that you can make in `occnp_custom_values_24.3.0.yaml` file to configure service and container ports.

> **Note**
>
> For upgrade scenario, changing port will cause temporary service disruption.

To override the default port numbers, used by service and container ports, and customize them as per your requirements, you can configure the following configurable parameters in custom-values.yaml file:

Table 3-61 Customizable Parameters for Service Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.servicePorts.pcfAmServiceHttp	HTTP signaling port for AM service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.pcfAmServiceHttps	HTTP signaling port for AM service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.bulwarkServiceHttp	HTTP signaling port for Bulwark service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.15.0	
global.servicePorts.appInfoHttp	HTTP signaling port for app info .	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcAppInfoHttp`
global.servicePorts.auditServiceHttp	HTTP signaling port for audit service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.bindingHttp	HTTP signaling port for binding service.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	
global.servicePorts.bindingHttps	HTTPS signaling port for binding service.	Optional	9443	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	
global.servicePorts.cmServiceHttp	HTTP signaling port for CM service.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	

Table 3-61 (Cont.) Customizable Parameters for Service Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.servicePorts.configServerHttp	HTTP signaling port for config server.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	Same value as `svcConfigServerHttp`
global.servicePorts.diamConnectorHttp	HTTP signaling port for Diameter connector.	Optional	8000	CNCPolicy & PCF	Updated in Release 1.8.1	The name for this parameter has been updated from `pcfDiamConnectorHttp` to `diamConnectorHttp`.
global.servicePorts.diamConnectorDiameter	Port for Diameter connector.	Optional	3868	CNCPolicy & PCF	Updated in Release 1.8.1	The name for this parameter has been updated from `pcfDiamConnectorDiameter` to `diamConnectorDiameter`.
global.servicePorts.ldapGatewayHttp	HTTP signaling port for LDAP Gateway.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	
global.servicePorts.ldapGatewayHttps	HTTPS signaling port for LDAP Gateway.	Optional	9443	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	

Table 3-61 (Cont.) Customizable Parameters for Service Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.servicePorts.diamGatewayHttp	HTTP signaling port for Diameter gateway.	Optional	8000	CNCPolicy & PCF	Updated in Release 1.8.1	The name for this parameter has been updated from `pcfDiamGatewayHttp` to `diamGatewayHttp`.
global.servicePorts.diamGatewayDiameter	Port for Diameter gateway.	Optional	3868	CNCPolicy & PCF	Updated in Release 1.8.1	The name for this parameter has been updated from `pcfDiamGatewayDiameter` to `diamGatewayDiameter`.
global.servicePorts.pcrfCoreDiameter	Port for PCRF Core Diameter.	Optional	3868	CNCPolicy & cnPCRF	Added in Release 1.7.3	
global.servicePorts.pcrfCoreHttp	HTTP signaling port for PCRF core service.	Optional	8000	CNCPolicy & cnPCRF	Added in Release 1.7.3	
global.servicePorts.pcrfDiamGatewayHttp	HTTP signaling port for PCRF Diameter Gateway.	Optional	8080	CNCPolicy & cnPCRF	Deprecated in Release 1.8.1	
global.servicePorts.pcrfDiamGatewayDiameter	Port for PCRF Diameter connector.	Optional	3868	CNCPolicy & cnPCRF	Deprecated in Release 1.8.1	
global.servicePorts.perfInfoHttp	HTTP signaling port for perf info.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcPerfInfoHttp`

Table 3-61 (Cont.) Customizable Parameters for Service Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.servicePorts.policydsHttp	HTTP signaling port for policyds.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	
global.servicePorts.preServiceHttp	HTTP signaling port for pre service.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	
global.servicePorts.preTestHttp	HTTP signaling port for pre test.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	
global.servicePorts.queryServiceHttp	HTTP signaling port for queryservice.	Optional	8000	CNCPolicy, PCF, & cnPCRF	Added in Release 1.7.3	
global.servicePorts.pcfSmServiceHttp	HTTP signaling port for SM service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.pcfSmServiceHttps	HTTPS signaling port for SM service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.soapConnectorHttp	HTTP signaling port for Soap connector.	Optional	8000	CNCPolicy & cnPCRF	Added in Release 1.7.3	
global.servicePorts.pcfUeServiceHttp	HTTP signaling port for UE service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.pcfUeServiceHttps	HTTPS signaling port for UE service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.udrConnectorHttp	HTTP signaling port for UDR Connector.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.udrConnectorHttps	HTTPS signaling port for UDR Connector.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.chfConnectorHttp	HTTP signaling port for CHF Connector.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.chfConnectorHttps	HTTPS signaling port for CHF Connector.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	
global.servicePorts.ingressGatewayHttp	HTTP signaling port for Ingress Gateway.	Optional	8000	CNCPolicy & PCF	Added in Release 22.1.0	
global.servicePorts.egressGatewayHttp	HTTP signaling port for Egress Gateway.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcEgressGatewayHttp`

Table 3-61 (Cont.) Customizable Parameters for Service Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.servicePorts.nrfClientNfDiscoveryHttp	HTTP signaling port for NRF client discovery service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfDiscoveryHttp`
global.servicePorts.nrfClientNfManagementHttp	HTTP signaling port for NRF client management service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfManagementHttp`
global.servicePorts.nrfClientNfDiscoveryHttps	HTTPS signaling port for NRF client discovery service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfDiscoveryHttps`
global.servicePorts.nrfClientNfManagementHttps	HTTPS signaling port for NRF client management service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfManagementHttps`
global.servicePorts.alternateRouteServiceHttp	HTTP signaling port for alternate route service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.8.0	Same value as `svcAlternateRouteServiceHttp`
global.servicePorts.alternateRouteServiceHazelcast	HTTP signaling port for alternate route Hazelcast service.	Optional	8000	CNC Policy & PCF	Added in Release 1.8.0	Same value as `svcAlternateRouteServiceHazelcast`
global.servicePorts.notifierServiceHttp	HTTP signaling port for Notifier service.	Optional	8000	CNC Policy & PCF	Added in Release 22.2.0	

Table 3-61 (Cont.) Customizable Parameters for Service Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.servicePorts.usageMonServiceHttp	HTTP signaling port for Usage Monitoring service.	Optional	8000	CNC Policy & PCF	Added in Release 22.2.0	
global.servicePorts.usageMonServiceHttps	HTTPS signaling port for Usage Monitoring service.	Optional	8443	CNC Policy & PCF	Added in Release 22.2.0	

Here is a sample of service ports configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

```
servicePorts:
    pcfAmServiceHttp: 8000
    pcfAmServiceHttps: 9443
    bulwarkServiceHttp: 8000
    appInfoHttp: &svcAppInfoHttp 8000
    auditServiceHttp: 8000
    bindingHttp: 8000
    bindingHttps: 9443
    cmServiceHttp: &svcCmServiceHttp 8000
    configServerHttp: &svcConfigServerHttp 8000
    diamConnectorHttp: 8000
    diamConnectorDiameter: 3868
    ldapGatewayHttp: 8000
    ldapGatewayHttps: 9443
    diamGatewayHttp: &svcDiamGatewayHttp 8000
    diamGatewayDiameter: 3868
    pcrfCoreDiameter: 3868
    pcrfCoreHttp: 8000
    perfInfoHttp: &svcPerfInfoHttp 8000
    policydsHttp: 8000
    preServiceHttp: 8000
    preTestHttp: 8000
    queryServiceHttp: 8000
    pcfSmServiceHttp: 8000
    pcfSmServiceHttps: 9443
    soapConnectorHttp: 8000
    pcfUeServiceHttp: 8000
    pcfUeServiceHttps: 9443
    udrConnectorHttp: 8000
    udrConnectorHttps: 9443
    chfConnectorHttp: 8000
    chfConnectorHttps: 9443
    ingressGatewayHttp: &svcIngressGatewayHttp 80
    egressGatewayHttp: &svcEgressGatewayHttp 8000
    nrfClientNfDiscoveryHttp: &svcNrfClientNfDiscoveryHttp 8000
    nrfClientNfManagementHttp: &svcNrfClientNfManagementHttp 8000
```

```
nrfClientNfDiscoveryHttps: &svcNrfClientNfDiscoveryHttps 9443
nrfClientNfManagementHttps: &svcNrfClientNfManagementHttps 9443
alternateRouteServiceHttp: &svcAlternateRouteServiceHttp 8000
alternateRouteServiceHazelcast: &svcAlternateRouteServiceHazelcast 8000
notifierServiceHttp: 8000
usageMonServiceHttp: 8000
usageMonServiceHttps: 8443
```

Table 3-62 Customizable Parameters for Container Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.containerPorts.monitoringHttp	HTTP signaling port for monitoring.	Optional	9000	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	Same value as `containerMonitoringHttp`
global.containerPorts.pcfAmServiceHttp	HTTP signaling port for AM service.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.pcfAmServiceHttps	HTTPS signaling port for AM service.	Optional	9443	CNC Policy & PCF	Added in Release 1.7.3	
global.containerPorts.bulwarkServiceHttp	HTTP signaling port for Bulwark service.	Optional	8080	CNC Policy & PCF	Added in Release 1.15.0	
global.containerPorts.appInfoHttp	HTTP signaling port for app info.	Optional	5906	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.auditServiceHttp	HTTP signaling port for Auditservice.	Optional	8081	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.bindingHttp	HTTP signaling port for binding service.	Optional	8080	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.bindingHttps	HTTPS signaling port for binding service.	Optional	8443	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.cmServiceHttp	HTTP signaling port for CMservice.	Optional	5807	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.configServerHttp	HTTP signaling port for config server.	Optional	8001	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	

Table 3-62 (Cont.) Customizable Parameters for Container Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.containerPorts.diamConnectorHttp	HTTP signaling port for Diameter Connector.	Optional	8080	CNCPolicy & PCF	Updated in Release 1.8.1	The name for this parameter has been updated from `pcfDiamConnectorHttp` to `diamConnectorHttp`.
global.containerPorts.diamConnectorDiameter	Diameter signaling port for Diam Connector.	Optional	3868	CNCPolicy & PCF	Updated in Release 1.8.1	The name for this parameter has been updated from `pcfDiamConnectorDiameter` to `diamConnectorDiameter`.
global.containerPorts.ldapGatewayHttp	HTTP signaling port for IDAP Gateway.	Optional	8084	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.diamGatewayHttp	HTTP signaling port for Diameter Gateway.	Optional	8080	CNCPolicy & PCF	Updated in Release 1.8.1	This parameter name has been updated from `pcfDiamGatewayHttp` to `diamGatewayHttp`.

Table 3-62 (Cont.) Customizable Parameters for Container Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.containerPorts.diamGatewayDiameter	Diameter signaling port for Diam Gateway.	Optional	3868	CNCPolicy & PCF	Updated in Release 1.8.1	This parameter name has been updated from `pcfDiamGatewayDiameter` to `diamGatewayDiameter`.
global.containerPorts.pcrfCoreDiameter	Diameter signaling port for PCRF core.	Optional	3868	CNCPolicy & cnPCRF	Added in Release 1.7.3	
global.containerPorts.pcrfCoreHttp	HTTP signaling port for PCRF Core service.	Optional	9080	CNCPolicy & cnPCRF	Added in Release 1.7.3	
global.containerPorts.pcrfDiamGatewayHttp	HTTP signaling port for PCRF Diameter Gateway.	Optional	8080	CNCPolicy & cnPCRF	Deprecated in Release 1.8.1	
global.containerPorts.pcrfDiamGatewayDiameter	PCRF diameter gateway.	Optional	3868	CNCPolicy & cnPCRF	Deprecated in Release 1.8.1	
global.containerPorts.perfInfoHttp	HTTP signaling port for perf-info.	Optional	5905	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.policydsHttp	HTTP signaling port for policyds.	Optional	8080	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.preServiceHttp	HTTP signaling port for pre service.	Optional	5806	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.preTestHttp	HTTP signaling port for pre test.	Optional	5806	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.queryServiceHttp	HTTP signaling port for queryservice.	Optional	8081	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
global.containerPorts.pcfSmServiceHttp	HTTP signaling port for SM service.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.pcfSmServiceHttps	HTTPS signaling port for SM service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	

Table 3-62 (Cont.) Customizable Parameters for Container Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.containerPorts.soapConnectorHttp	HTTP signaling port for soap connector.	Optional	8082	CNCPolicy & cnPCRF	Added in Release 1.7.3	
global.containerPorts.pcfUeServiceHttp	HTTP signaling port for UE service.	Optional	8082	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.pcfUeServiceHttps	HTTPS signaling port for UE service.	Optional	8081	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.pcfUserServiceHttp	HTTP signaling port for User service.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.pcfUserServiceHttps	HTTPS signaling port for User service.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.udrConnectorHttp	HTTP signaling port for UDR Connector.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.udrConnectorHttps	HTTPS signaling port for UDR Connector.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.chfConnectorHttp	HTTP signaling port for CHF connector.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.chfConnectorHttps	HTTPS signaling port for CHF connector.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
global.containerPorts.nrfClientNfDiscoveryHttp	HTTP signaling port for NRF client discovery.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfDiscoveryHttp`
global.containerPorts.nrfClientNfManagementHttp	HTTP signaling port for NRF client management.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfManagementHttp`

Table 3-62 (Cont.) Customizable Parameters for Container Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.containerPorts.nrfClientNfDiscoveryHttps	HTTPS signaling port for NRF client discovery.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfDiscoveryHttps`
global.containerPorts.nrfClientNfManagementHttps	HTTPS signaling port for NRF client management.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfManagementHttps`
global.containerPorts.ingressGatewayHttp	HTTP signaling port for Ingress Gateway.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerIngressGatewayHttp`
global.containerPorts.ingressGatewayHttps	HTTPS signaling port for Ingress Gateway.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerIngressGatewayHttps`

Table 3-62 (Cont.) Customizable Parameters for Container Ports Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.containerPorts.alternateRouteServiceHttp	HTTP signaling port for alternate route service.	Optional	8004	CNC Policy & PCF	Added in Release 1.8.0	Same value as `containerAlternateRouteServiceHttp`. This port configuration shall not be same as **alternateRouteServiceHazelcast** ,that is 8000, in this sample custom value file.
global.containerPorts.notifierServiceHttp	HTTP signaling port for Notifier service.	Optional	8080	CNC Policy & PCF	Added in Release 22.2.0	
global.containerPorts.usageMonServiceHttp	HTTP signaling port for Usage Monitoring service.	Optional	8000	CNC Policy & PCF	Added in Release 22.2.0	
global.containerPorts.usageMonServiceHttps	HTTPS signaling port for Usage Monitoring service.	Optional	8443	CNC Policy & PCF	Added in Release 22.2.0	

Here is a sample of service ports configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

```
containerPorts:
    monitoringHttp: &containerMonitoringHttp 9000
    pcfAmServiceHttp: 8000
    pcfAmServiceHttps: 9443
    bulwarkServiceHttp: 8080
    appInfoHttp: 8000
    auditServiceHttp: 8000
    bindingHttp: 8000
    bindingHttps: 9443
    cmServiceHttp: 8000
```

```
configServerHttp: 8000
diamConnectorHttp: 8000
diamConnectorDiameter: 3868
ldapGatewayHttp: 8000
diamGatewayHttp: 8000
diamGatewayDiameter: 3868
pcrfCoreDiameter: 3868
pcrfCoreHttp: 8000
perfInfoHttp: 8000
policydsHttp: 8000
preServiceHttp: 8000
preTestHttp: 8000
queryServiceHttp: 8000
pcfSmServiceHttp: 8000
pcfSmServiceHttps: 9443
soapConnectorHttp: 8000
pcfUeServiceHttp: 8000
pcfUeServiceHttps: 9443
udrConnectorHttp: 8000
udrConnectorHttps: 9443
chfConnectorHttp: 8000
chfConnectorHttps: 9443
nrfClientNfDiscoveryHttp: &containerNrfClientNfDiscoveryHttp 8000
nrfClientNfManagementHttp: &containerNrfClientNfManagementHttp 8000
nrfClientNfDiscoveryHttps: &containerNrfClientNfDiscoveryHttps 9443
nrfClientNfManagementHttps: &containerNrfClientNfManagementHttps 9443
ingressGatewayHttp: &containerIngressGatewayHttp 8000
ingressGatewayHttps: &containerIngressGatewayHttps 9443
alternateRouteServiceHttp: &containerAlternateRouteServiceHttp 8004
notifierServiceHttp: 8080
usageMonServiceHttp: 8000
usageMonServiceHttps: 8443
```

Table 3-63 Customizable Parameters for Ports Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.publicHttpSignalingPort	HTTP/2.0 Port of ingress gateway	Optional	80	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	If `httpsEnabled` is set to false, this Port would be HTTP/2.0 Port (unsecured).

Table 3-63 (Cont.) Customizable Parameters for Ports Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.publicHttpsSignallingPort	HTTPS/2.0 Port of ingress gateway	Optional	443	CNC Policy, PCF, &cnPCRF	Deprecated in Release 1.14.0	Set this parameter to 0 if HTTPS is disabled.
global.publicHttpsSignalingPort	HTTPS/2.0 Port of ingress gateway	Optional	443	CNC Policy, PCF, &cnPCRF	Added in Release 1.14.0	If `httpsEnabled` is set to true, this Port would be HTTPS/ 2.0 port (secured SSL).
global.configServerPort	HTTP signaling port for config server.	Optional	5807	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.3	same vale as `svcConfigServerHttp`

Table 3-63 (Cont.) Customizable Parameters for Ports Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.ports.actuatorPort	Actuator Port	Optional	Optional	*containerMonitoringHttp	CNCPolicy , PCF, &cnPCRF	Added in Release 1.8.0
ingress-gateway.ports.containerPort	Container Port represents a network port in a single container	Optional	*containerIngressGatewayHttp	CNCPolicy , PCF, &cnPCRF	Added in Release 1.8.0	Same value as `containerIngressGatewayHttp`
ingress-gateway.ports.containersslPort	Container Port represents a network ssl port in a single container	Optional	*containerIngressGatewayHttps	CNCPolicy , PCF, &cnPCRF	Added in Release 1.8.0	Same value as `containerIngressGatewayHttps`

Same value as containerMonitoringHttp

Here is a sample of configurable parameters for ingress-gateway's ports in `occnp_custom_values_24.3.0.yaml` file:

```
# -----Ingress Gateway Settings - BEGIN-----
  # If httpsEnabled is false, this Port would be HTTP/2.0 Port (unsecured)
  publicHttpSignalingPort: 80
  # If httpsEnabled is true, this Port would be HTTPS/2.0 Port (secured SSL)
  publicHttpsSignallingPort: 443
  configServerPort: *svcConfigServerHttp



ingress-gateway:
  ports:
    actuatorPort: *containerMonitoringHttp
    containerPort: *containerIngressGatewayHttp
    containersslPort: *containerIngressGatewayHttps
```

Table 3-64 Customizable Parameters for Ports Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.serviceEgressGateway.actuatorPort	Actuator Port	Optional	*containerMonitoringHttp	CNCPolicy & PCF	Added in Release 1.8.0	Same value as `containerMonitoringHttp`
egress-gateway.serviceEgressGateway.Port	Service EgressGateway port	Optional	*svcEgressGatewayHttp	CNCPolicy , PCF, &cnPCRF	Added in Release 1.8.0	Same value as `svcEgressGatewayHttp`

Here is a sample of configurable parameters for egress-gateway's ports in `occnp_custom_values_24.3.0.yaml` file:

```
egress-gateway:
  serviceEgressGateway:
    actuatorPort: *containerMonitoringHttp
    port: *svcEgressGatewayHttp
```

Table 3-65 Customizable Parameters for Ports Configuration in nrf-client-nfdiscovery

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.nrf-client-nfdiscovery.envPlatformServicePort	HTTP signaling port for app info.	Optional	5906	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcAppInfoHttp`
global.nrf-client-nfdiscovery.envPerformanceServicePort	HTTP signaling port for perf info.	Optional	5905	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcPerfInfoHttp`
global.nrf-client-nfdiscovery.envCfgServerPort	HTTP signaling port for config server.	No	5807	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.3	same vale as `svcConfigServerHttp`
global.nrf-client-nfdiscovery.containerHttpPort	HTTP signaling port for NRF client discovery.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfDiscoveryHttp`
global.nrf-client-nfdiscovery.containerHttpsPort	HTTPS signaling port for NRF client discovery.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfDiscoveryHttps`
global.nrf-client-nfdiscovery.serviceHttpPort	HTTP signaling port for NRF client discovery service.	Optional	5910	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfDiscoveryHttp`
global.nrf-client-nfdiscovery.serviceHttpsPort	HTTPS signaling port for NRF client discovery service.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfDiscoveryHttps`

Here is a sample of configurable parameters for nrf-client-nfdiscovery's ports in `occnp_custom_values_24.3.0.yaml` file:

```
nrf-client-nfdiscovery:
    envPlatformServicePort: *svcAppInfoHttp
    envPerformanceServicePort: *svcPerfInfoHttp
    envCfgServerPort: *svcConfigServerHttp
    containerHttpPort: *containerNrfClientNfDiscoveryHttp
    containerHttpsPort: *containerNrfClientNfDiscoveryHttps
    serviceHttpPort: *svcNrfClientNfDiscoveryHttp
    serviceHttpsPort: *svcNrfClientNfDiscoveryHttps
```

Table 3-66 Customizable Parameters for Ports Configuration in nrf-client-nfmanagement

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.nrf-client-nfmanagement.envPlatformServicePort	HTTP signaling port for app info.	Optional	5906	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcAppInfoHttp`
global.nrf-client-nfmanagement.envPerformanceServicePort	HTTP signaling port for perf info.	Optional	5905	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcPerfInfoHttp`
global.nrf-client-nfmanagement.envCfgServerPort	HTTP signaling port for config server.	Optional	5807	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.3	same vale as `svcConfigServerHttp`
global.nrf-client-nfmanagement.containerHttpPort	HTTP signaling port for NRF client discovery.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfManagementHttp`
global.nrf-client-nfmanagement.containerHttpsPort	HTTPS signaling port for NRF client discovery.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfManagementHttps`

Table 3-66 (Cont.) Customizable Parameters for Ports Configuration in nrf-client-nfmanagement

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.nrf-client-nfmanagement.serviceHttpPort	HTTP signaling port for NRF client discovery service.	Optional	5910	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfManagementHttp`
global.nrf-client-nfmanagement.serviceHttpsPort	HTTPS signaling port for NRF client discovery service.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfManagementHttps`

Here is a sample of configurable parameters for nrf-client-nfmanagement's ports in `occnp_custom_values_24.3.0.yaml` file:

```
nrf-client-nfmanagement:
    envPlatformServicePort: *svcAppInfoHttp
    envPerformanceServicePort: *svcPerfInfoHttp
    envCfgServerPort: *svcConfigServerHttp
    containerHttpPort: *containerNrfClientNfManagementHttp
    containerHttpsPort: *containerNrfClientNfManagementHttps
    serviceHttpPort: *svcNrfClientNfManagementHttp
    serviceHttpsPort: *svcNrfClientNfManagementHttps
```

Table 3-67 Customizable Parameters for Ports Configuration in Alternate Route Service

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.ports.servicePort	HTTP signaling port for alternate route service.	Optional	8000	CNCPolicy & PCF	Added in Release 1.8.0	Same value as `svcAlternateRouteServiceHttp`

Table 3-67 (Cont.) Customizable Parameters for Ports Configuration in Alternate Route Service

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.ports.containerPort	HTTP signaling port for alternate route service.	Optional	8004	CNCPolicy & PCF	Added in Release 1.8.0	Same value as `containerAlternateRouteServiceHttp`
alternate-route.ports.actuatorPort	HTTP signaling port for monitoring.	Optional	9000	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	Same value as `containerMonitoringHttp`
alternate-route.hazelcast.port	HTTP signaling port for alternate route's Hazlecast .	Optional	8000	CNCPolicy & PCF	Added in Release 1.8.0	Same value as `svcAlternateRouteServiceHazelcast`

Here is a sample of configurable parameters for alternate route service's ports in `occnp_custom_values_24.3.0.yaml` file:

```
alternate-route:
  ports:
    servicePort: *svcAlternateRouteServiceHttp
    containerPort: *containerAlternateRouteServiceHttp
    actuatorPort: *containerMonitoringHttp
  hazelcast:
    port: *svcAlternateRouteServiceHazelcast
```

3.22 Aspen Service Mesh Configurations

This section describes the customizatons that you can make in `occnp_custom_values_24.3.0.yaml` files to configure Aspen Service Mesh (ASM) in the Oracle Communications Cloud Native Core Policy.

1. Enable ASM by setting the value for `serviceMeshEnabled` parameter, under global section, as true.
2. Configure the values for the parameters described in the following table:

Table 3-68 Configurable Parameters for Aspen Servicemesh Configuration

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
istioSidecarQuitUrl	Specifies quit URL that can be configured for side car.	Conditional	`http://127.0.0.1:15000/quitquitquit`	CNC Policy & PCF	Added in Release 1.10.2	Applicable only when `serviceMeshEnabled` parameter is set to true.
istioSidecarReadyUrl	Specifies readiness URL that can be configured for side car.	Conditional	`http://127.0.0.1:15000/ready`	CNC Policy & PCF	Added in Release 1.10.2	Applicable only when `serviceMeshEnabled` parameter is set to true.

3. In the global section, uncomment the following annotations to include port 9000 - a Prometheus scrap port

```
allResources:
      labels: {}
      annotations: {
      #Enable this section for service-mesh based installation
        # traffic.sidecar.istio.io/excludeInboundPorts: "9000",
        # traffic.sidecar.istio.io/excludeOutboundPorts: "9000"
```

4. (Optional) If CNC Policy is deployed with OSO, the pods need to have an annotation **oracle.com/cnc: true**.

```
customExtension:
    # The `factoryLabelTemplates` and `factoryAnnotationTemplates` can
    # accept templates rather than plain text.
    factoryLabelTemplates: {}
    factoryAnnotationTemplates: {}

    allResources:
      labels: {}
      annotations:
          sidecar.istio.io/inject: "false"

    lbServices:
      labels: {}
      annotations:
          oracle.com/cnc: "true"

    lbDeployments:
      labels: {}
      annotations:
          oracle.com/cnc: "true"
```

```
        sidecar.istio.io/inject: "true"

  nonlbServices:
    labels: {}
    annotations:
        oracle.com/cnc: "true"

  nonlbDeployments:
    labels: {}
    annotations:
        oracle.com/cnc: "true"
        sidecar.istio.io/inject: "true"
```

5. Uncomment the following annotations in the deployment sections of following services in their deployment sections:
   - `nrf-client-nfdiscovery.nrf-client-nfmanagement`
   - `ingress-gateway`
   - `egress-gateway`
   - `alternate-route`
   - `bulwark`

```
deployment:
      customExtension:
        annotations: {
           #Enable this section for service-mesh based
installation:
  #           traffic.sidecar.istio.io/excludeOutboundPorts:
"9000,8095,8096,7,53",
  #           traffic.sidecar.istio.io/excludeInboundPorts:
"9000,8095,8096,7,53"
        }
```

   Here, 8095 and 8096 are Coherence ports.

   > **Note**
   >
   > Port 53 is included only if DNS lookup bypasses the sidecar connection management.

6. Uncomment the following annotations in the deployment sections of `diam-gateway` service:

```
deployment:
      customExtension:
        annotations: {
           #Enable this section for service-mesh based
installation:
  #           traffic.sidecar.istio.io/excludeOutboundPorts: "9000,5801,7",
  #           traffic.sidecar.istio.io/excludeInboundPorts: "9000,5801,7"
      }
```

7. **Disable init containers**: Init containers do not work when the namespace has istio or aspen service mTLS enabled. To disable init containers, set the value for `initContainerEnable` to false in custom values file.

```
global:
  initContainerEnable: false
```

3.23 OAUTH Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` files to configure OAUTH in Ingress and Egress Gateway.

> **Note**
>
> These configurations are applicable when the Ingress Gateway and Egress Gatway are enabled and the NRF Client services are enabled.

To configure OAUTH in ingress-gateway, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-69 Configurable Parameters for OAUTH Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional / Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.oauthValidatorEnabled	Enable or disable the OAuth Validator.	Optional	false	CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.nfInstanceId	NF Instance Id of the service producer.	Optional	6faf1bbc-6e4a-4454-a507-a14ef8e1bc11	CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.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 by JWT.	Optional	0	CNC Policy & PCF	Added in Release 1.6.x	
ingress-gateway.nrfPublicKeyKubeSecret	Name of the secret which stores the public key(s) of NRF	Optional		CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.nrfPublicKeyKubeNamespace	Namespace of the NRF public key secret	Optional		CNC Policy & PCF	Added in Release 1.5.x	

Table 3-69 (Cont.) Configurable Parameters for OAUTH Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional / Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.validationType	Possible values are: • strict • relaxed strict - If incoming request does not contain "Authorization" (Access Token) header, the request is rejected. relaxed - If incoming request contains "Authorization" header, it is validated. If incoming request does not contain "Authorization" header, validation is ignored.	Optional	relaxed	CNC Policy & PCF	Added in Release 1.6.x	
ingress-gateway.producerPlmnMNC	MNC of the service producer	Optional	123	CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.producerPlmnMCC	MCC of the service producer	Optional	456	CNC Policy & PCF	Added in Release 1.5.x	

Table 3-69 (Cont.) Configurable Parameters for OAUTH Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional / Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.producerScope	Contains the NF service name(s) of the NF service producer(s). The service name(s) included in this attribute shall be any of the services defined in the ServiceName enumerated type. **Note**: `producerScope` must be configured in `custom-values.yaml` only if different from the default values.	Mandatory	npcf-smpolicycontrol, npcf-ampolicycontrol, npcf-uepolicycontrol, npcf-policyauthorization	CNC Policy & PCF	Added in Release 1.12.0	

Here is a sample OAUTH configurations in ingress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
----OAUTH CONFIGURATION - BEGIN ----
oauthValidatorEnabled: false
nfInstanceId: 6faf1bbc-6e4a-4454-a507-a14ef8e1bc11
allowedClockSkewSeconds: 0
nrfPublicKeyKubeSecret: ''
nrfPublicKeyKubeNamespace: ''
validationType:
relaxed producerPlmnMNC: 123
producerPlmnMCC: 456
producerScope: npcf-smpolicycontrol, npcf-ampolicycontrol, npcf-
uepolicycontrol, npcf-policyauthorization
  # ----OAUTH CONFIGURATION - END ----
```

Table 3-70 Configurable Parameters for OAUTH Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional/ Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.oauthClient.enabled	Determines if the oAuthClient lookup is enabled or not (static configuration)	Optional	false	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.oauthClient.dnsSrvEnabled	Enable/Dsiable the DNS-SRV query to coreDNS Server	Optional	false	CNC Policy& PCF	Removed in Release 1.12.0	
egress-gateway.oauthClient.nrfClientQueryEnabled	Determines if NRF-Client Query is enabled or not (Dynamic configuration).	Optional	false	CNC Policy& PCF	Added in Release 1.11.0	
egress-gateway.oauthClient.httpsEnabled	Determines if https support is enabled or not which is a deciding factor for oauth request scheme.	Optional	false	CNC Policy& PCF	Added in Release 1.8.0	
egress-gateway.oauthClient.virtualFqdn	Indicates the name of the virtual FQDN or FQDN that needs to be sent in the dns-srv query.	Conditional (If dnsSrvEnabled is set to true.)	string Example: nrf.oracle.com:80	CNC Policy& PCF	Added in Release 1.8.0	
egress-gateway.oauthClient.staticNrfList	List of Static NRF instances that need to be used for oAuth requests when `nrfClientQueryEnabled` is false.	Conditional (If oAuth is enabled.)		CNC Policy& PCF	Added in Release 1.8.0	
egress-gateway.oauthClient.nfType	NFType of service consumer.	Conditional (If oAuth is enabled.)		CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.oauthClient.nfInstanceId	NF InstanceId of service consumer.	Optional	fe7d992b-0541-4c7d-ab84-c6d70b1b01b1	CNC Policy& PCF	Added in Release 1.5.x	Modify the parameter with actual value, if OAuth is enabled.

Table 3-70 (Cont.) Configurable Parameters for OAUTH Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional/ Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.oauthClient.consumerPlmnMNC	MNC of service Consumer	Optional	345	CNC Policy& PCF	Added in Release 1.5.x	Modify the parameter with actual value, if OAuth is enabled.
egress-gateway.oauthClient.consumerPlmnMCC	MCC of service Consumer	Optional	567	CNC Policy& PCF	Added in Release 1.5.x	Modify the parameter with actual value, if OAuth is enabled.
egress-gateway.oauthClient.maxRetry	Maximum number of retry that need to be performed to other NRF Fqdn's in case of failure response from first contacted NRF based on the errorCodeSeries configured.	Conditional (If oAuth is enabled.)	2	CNC Policy& PCF	Added in Release 1.8.0	
egress-gateway.oauthClient.apiPrefix	apiPrefix that needs to be appended in the Oauth request flow while sending AccessToken requests to NRF instances.	Conditional (If oAuth is enabled.)	""	CNC Policy& PCF	Added in Release 1.8.0	

Table 3-70 (Cont.) Configurable Parameters for OAUTH Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional/ Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.oauthClient.errorCodeSeries	Determines the fallback condition to other non primary NRF instances if the attempts configured for the current NRF instance in use are exhausted and if the last received response from NRF matches configured value of `retryErrorCodeSeries` for any errorSetId (4XX, 5XX).	Conditional (If oAuth is enabled and required a different error code series.)	4XX	CNC Policy& PCF	Added in Release 1.8.0	
egress-gateway.oauthClient.retryAfter	RetryAfter value in milliseconds that needs to be set for a particular NRF Fqdn. If a retryAfter value is received from a particular NRF instance then irrespective of attempts for primary/ non-primary NRF instances count and retryErrorCodeSeries configurations at EGW, fallback to an alternate non-primary NRF instance based on its availability and priority takes place.	Conditional (If oAuth is enabled.)	5000	CNC Policy & PCF	Added in Release 1.8.0	

Table 3-70 (Cont.) Configurable Parameters for OAUTH Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional/ Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.oauthClient.nrfClientConfig	Determines the NRF-Client Mgmt Svc configurations which are required when dynamic configurations are in place at Egress-Gateway.	Optional		CNC Policy & PCF	Added in Release 1.11.0	
egress-gateway.oauthClient.nrfClientConfig.serviceName	The service name of NRF-Client Mgmt Svc.	Optional	occnp-nrf-client-nfmanagement	CNC Policy & PCF	Added in Release 1.11.0	
egress-gateway.oauthClient.nrfClientConfig.host	The address of NRF-Client Mgmt Svc	Optional	10.233.49.44	CNC Policy & PCF	Added in Release 1.11.0	
egress-gateway.oauthClient.nrfClientConfig.port	Determines the port configuration for NRF-Client Mgmt Svc for sending Subscription requests.	Optional	8000	CNC Policy & PCF	Added in Release 1.11.0	
egress-gateway.oauthClient.nrfClientRequestMap	Determines the request mapping URL for sending Subscription requests from Egress-Gateway to NRF-Client Mgmt Svc.	Optional	/v1/nrf-client/subscriptions/nrfRouteList	CNC Policy & PCF	Added in Release 1.11.0	

Table 3-70 (Cont.) Configurable Parameters for OAUTH Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional/ Conditional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.oauthClient.oauthDeltaExpiryTime	Determines the lifespan of the received tokens. This flag has default value of 0 millisecond. This value gets reduced from the TTL as received from NRF when calculating the lifespan of a received token. Here, the token is saved in the coherence cache of the Egress Gateway pod and expires after 55 seconds, so any requests after this duration requires a new token fetch and thus avoiding expired token usage.	Optional	0	CNC Policy & PCF	Added in Release 22.2.0	The duration can be fine tuned depending upon TTL. For Example: When TTL is 60 secs, then `oauthDeltaExpiryTime` can be set to fine tune the token fetch duration to 55 sec. A range of 3 to 7 seconds depending upon the TTL.

Here is a sample OAUTH configurations in egress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
# ---- Oauth Configuration - BEGIN ----
  oauthClient:
    enabled: false
    dnsSrvEnabled: false
    nrfClientQueryEnabled: false
    httpsEnabled: false
    virtualFqdn: nrf.oracle.com:80
    staticNrfList:
      - nrf1.oracle.com:80
```

```
  nfInstanceId: fe7d992b-0541-4c7d-ab84-c6d70b1b01b1
  consumerPlmnMNC: 345
  consumerPlmnMCC: 567
  maxRetry: 2
  apiPrefix: ""
  errorCodeSeries: 4XX
  retryAfter: 5000
  nrfClientConfig:
    serviceName: "occnp-nrf-client-nfmanagement"
    host: 10.233.49.44
    port: 8000
    nrfClientRequestMap: "/v1/nrf-client/subscriptions/nrfRouteList"
# ---- Oauth Configuration - END ----
```

Authorization Request for Producer NFs

This section provides information on how to enable or disable sending `oc-access-token-request-info` header in the outgoing requests. When this parameter is set to NONE, PCF does not request the authorization token to any service and OAuth validation at the producer NF's Ingress Gateway depends on OAuth validator Configurations.

The following table describes the parameters that users can customize to enable or disable authorization for producer network functions:

> **Note**
>
> The default configuration value can be changed only when OAuth client is enabled at Egress Gateway.

Table 3-71 Configurable Parameters for OAUTH Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional/ Conditional Parameter	Default Value	Applicable to Deployment
sm-service.envOathAccessTokenType	Specifies whether to skip or send the authorization portion of packages sent out from Egress Gateway when requesting OAuth2 tokens. When the value is set to NONE, the header is skipped, and not pegged to the outgoing packages when communicating with other NFs. When the value is set to **NF_TYPE**, the header is included in the outgoing request and `targetNfType` is set to the corresponding NF. When the value is set to **NF_INSTANCE_ID**, the header is included in the outgoing request and `targetNfInstanceId` is set to the corresponding Instance ID of producer NF.	Optional	NONE Possible values are: • NONE • NF_TYPE • NF_INSTANCE_ID NF_TYPE and NF_INSTANCE_ID will be set if the Oauth Client is enabled at Egress Gateway.	PCF
user-service.envOathAccessTokenTypeUdr	Specifies whether to skip or send the authorization portion of packages, sent out from Egress Gateway towards UDR, when requesting OAuth2 tokens. When the value is set to NONE, the header is skipped, and not pegged to the outgoing packages when communicating with other NFs. When the value is set to **NF_TYPE**, the header is included in the outgoing request and `targetNfType` is set to the corresponding NF. When the value is set to **NF_INSTANCE_ID**, the header is included in the outgoing request and `targetNfInstanceId` is set to the corresponding Instance ID of producer NF.	Optional	NONE Possible Values: • NONE • NF_TYPE • NF_INSTANCE_ID NF_TYPE and NF_INSTANCE_ID will be set if the Oauth Client is enabled at Egress Gateway.	PCF

Table 3-71 (Cont.) Configurable Parameters for OAUTH Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional/ Conditional Parameter	Default Value	Applicable to Deployment
user-service.envOathAccessTokenTypeChf	Specifies whether to skip or send the authorization portion of packages, sent out from Egress Gateway towards CHF, when requesting OAuth2 tokens. When the value is set to NONE, the header is skipped, and not pegged to the outgoing packages when communicating with other NFs. When the value is set to **NF_TYPE**, the header is included in the outgoing request and `targetNfType` is set to the corresponding NF. When the value is set to **NF_INSTANCE_ID**, the header is included in the outgoing request and `targetNfInstanceId` is set to the corresponding Instance ID of producer NF.	Optional	NONE Possible Values: • NONE • NF_TYPE • NF_INSTANCE_ID NF_TYPE and NF_INSTANCE_ID will be set if the Oauth Client is enabled at Egress Gateway.	PCF
binding.envOathAccessTokenType	Specifies whether to skip or send the authorization portion of packages sent out from Egress Gateway when requesting OAuth2 tokens. When the value is set to NONE, the header is skipped, and not pegged to the outgoing packages when communicating with other NFs. When the value is set to **NF_TYPE**, the header is included in the outgoing request and `targetNfType` is set to the corresponding NF. When the value is set to **NF_INSTANCE_ID**, the header is included in the outgoing request and `targetNfInstanceId` is set to the corresponding Instance ID of producer NF.	Optional	NONE Possible Values: • NONE • NF_TYPE • NF_INSTANCE_ID NF_TYPE and NF_INSTANCE_ID will be set if the Oauth Client is enabled at Egress Gateway.	PCF

The following is the snippet of the `occnp_custom_values_24.3.0.yaml` file:

```
sm-service:
   envOathAccessTokenType: 'NONE'
user-service:
  envOathAccessTokenTypeUdr: 'NONE'
  envOathAccessTokenTypeChf: 'NONE'
binding:
  envMysqlDatabase: occnp_binding
  envOathAccessTokenType: 'NONE'
```

3.24 XFCC Header Validation Configuration

This section describes the customizatons that you can make in `occnp_custom_values_24.3.0.yaml` files to configure XFCC header.

XFCC introduces support for CNC Policy as a producer, to check, if Service Communication Proxy (SCP) which has sent the HTTP request is the same proxy consumer/client, which is expected to send a HTTP2 request. This is achieved by comparing the FQDN of the SCP present in the “x-forwarded-client-cert” (XFCC) of http2 header, with the FQDN of the SCPs configured in the CNC Policy.

For more information about the XFCC header, see *Oracle Communications Cloud Native Core Policy User's Guide*.

To configure XFCC header, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-72 Configurable Parameters for XFCC Header Validation Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.xfccHeaderValidation.validation.enabled	Determines if the incoming XFCC header needs to be validated.	Optional	false	CNCPolicy & PCF	Added in Release 1.8.0	
ingress-gateway.xfccHeaderValidation.validation.peerList	Specifies the list of configured NF FQDN’s against which the matchField entry configured, present in the XFCC Header will be validated.	Conditional (If xfccHeader validation is enabled.)		CNC Policy & PCF	Updated in Release 22.1.0	

Table 3-72 (Cont.) Configurable Parameters for XFCC Header Validation Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.xfccHeaderValidation.validation.matchCerts	Specifies the number of certificates that need to be validated starting from the right most entry in the XFCC header. • If the parameter is set to -1, validation is performed against all entries. Click here for Example. • If parameter is set to a positive number, validation is performed from starting from the right to left. In case value is set to 2,the two right most entries will be validated to find a match. Click here for Example.	Conditional (If xfccHeader validation is enabled.)	-1	CNC Policy & PCF	Added in Release 1.8.0	Note: If there are multiple certificates defined in XFCC header, all the entries are validated from the right to left till a match is found. If the match is found, the Ingress Gatway stops and forwards the response to backend microservice. If no match is found, 400 Bad Request is returned as a response from Ingress Gateway.

Table 3-72 (Cont.) Configurable Parameters for XFCC Header Validation Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.xfccHeaderValidation.validation.matchField	Specifies a field in a corresponding XFCC header against which the configured nfList FQDN validation needs to be performed.	Conditional (If xfccHeader validation is enabled.)	DNS	CNC Policy & PCF	Added in Release 1.8.0	Note: If there are multiple DNS entries defined in XFCC header, all the entries are validated from the right to left till a match is found. Click here for Example. If the match is found, the Ingress Gatway stops and forwards the response to backend microservice. If no match is found, 400 Bad Request is returned as a response from Ingress Gateway.
ingress-gateway.xfccHeaderValidation.validation.dnsResolutionInterval	Specifies the interval (in milliseconds) used to resolve failed FQDNs.	Optional	300000	CNC Policy & PCF	Added in CNC Policy 22.1.0	
global.xfccHeaderValidation.validation.errorTrigger[i].exceptionType	Specifies the configurable exception or error type for an error scenario in Ingress Gateway.	Optional	XFCC_HEADER_INVALID XFCC_MATCHCERTCOUNT_GREATER_THAN_CERTS_IN_HEADER XFCC_HEADER_NOT_PRESENT_OR_EMPTY	CNC Policy & PCF	Added in CNC Policy 22.1.0	

Table 3-72 (Cont.) Configurable Parameters for XFCC Header Validation Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.xfccHeaderValidation.validation.errorTrigger[i].errorCode	Specifies the configurable error code to be returned when the exception or error configured in exceptionType occurs at Ingress Gateway.	Optional	401 402 403	CNC Policy & PCF	Added in CNC Policy 22.1.0	
global.xfccHeaderValidation.validation.errorTrigger[i].errorCause	Specifies the configurable error cause to be returned when the exception or error configured in exceptionType occurs at Ingress Gateway.	Optional	xfcc header is invalid matchCerts count is greater than the certs in the request xfcc header is not present or empty in the request	CNC Policy & PCF	Added in CNC Policy 22.1.0	
global.xfccHeaderValidation.validation.errorTrigger[i].errorTitle	Specifies the configurable error title to be returned when the exception or error configured in exceptionType occurs at Ingress Gateway.	Optional	Invalid XFCC Header	CNC Policy & PCF	Added in CNC Policy 22.1.0	
global.xfccHeaderValidation.validation.errorTrigger[i].errorDescription	Specifies the configurable error description to be returned when the exception or error configured in exceptionType occurs at Ingress Gateway.	Optional	Invalid XFCC Header	CNC Policy & PCF	Added in CNC Policy 22.1.0	

If the **ingressgateway.xfccHeaderValidation.validation.matchCerts** parameter is set to -1, validation to be performed against all entries. All the entries written in bold are validated till the match is found.

By=http://router1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8de68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=scp1.com"; URI=http://testenv1.blr.com; DNS=scp8.com;DNS=scp1.com; DNS=scp6.com, By=http://router1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8de68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=scp10.com"; URI=http://testenv1.blr.com; DNS=scp10.com; DNS=scp8.com; DNS=scp9.com, By=http://routexr1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8de68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=scp4.com"; URI=http://testenv1.blr.com; DNS=scp9.com; DNS=scp4.com;DNS=scp1.com

If the **ingressgateway.xfccHeaderValidation.validation.matchCerts** parameter is set to 2, the two right most entries, written in bold, are validated to find a match.

By=http://router1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8de68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=scp10.com"; URI=http://testenv1.blr.com; DNS=scp10.com; DNS=scp8.com; DNS=scp9.com, By=http://routexr1.blr.com;Hash=468ed33be74eee6556d90c0149c1309e9ba61d6425303443c0748a02dd8de68; Subject="/C=US/ST=CA/L=San Francisco/OU=Lyft/CN=scp4.com"; URI=http://testenv1.blr.com; DNS=scp9.com; DNS=scp4.com;DNS=scp1.com

If the **ingress-gateway.xfccHeaderValida tion.validation.matchField** parameter has multiple DNS entries, all entries are validated till a match is found.

The following is a sample snippet of XFCC Header configurations under ingress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
global:
    xfccHeaderValidation:
      validation:
        enabled: false
        peerList:
          - name: scp.com
          - name: smf.com
          - name: amf.com
          - name: scp1.com
            enabled: true
          - name: scp2.com
          - name: scp3.com
            enabled: false
          - name: xyz.test.com
            enabled: true
            scheme: http
            type: virtual
          - name: abc.test.com
            enabled: true
            scheme: https
            type: virtual
          - name: xfcc.test.com
            enabled: false
            scheme: http
            type: virtual
        matchCerts: -1
        matchField: DNS
```

```
        dnsResolutionInterval: 300000
```

XFCC Header - Route Level

To enable or disable XFCC header per route, set the `validationEnabled` parameter to true under each route (in Ingress Gateway):

```
routesConfig:
    - id: sm_create_session_route
      uri: http://{{ .Release.Name }}-occnp-pcf-sm:
{{ .Values.global.servicePorts.pcfSmServiceHttp }}
      path: /npcf-smpolicycontrol/*/sm-policies
      order: 1
      method: POST
      readBodyForLog: true
      filters:
        subLog: true,CREATE,SM
      metadata:
        xfccHeaderValidation:
          validationEnabled: false
```

> **Note**
>
> These routes are for internal consumption and determine how the incoming traffic is distributed among microservices on the basis of routing properties. To make any modification to these routes other than enabling or disabling XFCC header feature, kindly contact My Oracle Support.

3.25 Ingress/Egress Gateway HTTPS Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` files to configure HTTPS in ingress/egress gateway.

> **Note**
>
> These configurations are applicable only when ingress/egress gateway is enabled and the following parameters are set to true in custom-yaml file:
>
> - ingress-gateway.enableIncomingHttps
> - egress-gateway.enableOutgoingHttps

To configure HTTPS in ingress-gateway, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-73 Configurable Parameters for HTTPS Configurations in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.enableIncomingHttps	To enable https for ingress traffic	No	False	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	
ingress-gateway.service.ssl.privateKey.k8SecretName	Name of the Kubernetes Secret which contains the private key for Policy	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.privateKey.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the private key for Policy can be found	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.privateKey.rsa.fileName	rsa private key file name.	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.certificate.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.certificate.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.certificate.rsa.fileName	rsa private key file name	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.caBundle.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true

Table 3-73 (Cont.) Configurable Parameters for HTTPS Configurations in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.service.ssl.caBundle.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.caBundle.fileName	private key file name	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.keyStorePassword.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttp is true
ingress-gateway.service.ssl.keyStorePassword.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.keyStorePassword.fileName	File name that has password for keyStore	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.trustStorePassword.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.trustStorePassword.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true
ingress-gateway.service.ssl.trustStorePassword.fileName	File name that has password for trustStore	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true

Table 3-73 (Cont.) Configurable Parameters for HTTPS Configurations in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingressServer.keepAlive.enabled	If enabled nettyserver will send keep alive message for eachconnection	No	false		Added in Release 1.7.3	
ingressServer.keepAlive.idealTime	Time after which keep alive will be tried after successful response from the peer	No	180 (in seconds)		Added in Release 1.7.3	
ingressServer.keepAlive.count	Number of times it should retry if there is no response for keep alive	No	9		Added in Release 1.7.3	
ingressServer.keepAlive.interval	The interval after which it should retry in case of failure	No	60 (in seconds)		Added in Release 1.7.3	
global.configServerPort	The Configuration Server port	No	*svcConfigServerHttp	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.3	

Here is a sample HTTPS configurations in ingress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
# ---- HTTPS Configuration - BEGIN ----
  enableIncomingHttps: true

  service:
    ssl:
      privateKey:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        rsa:
          fileName: rsa_private_key_pkcs1.pem
      certificate:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        rsa:
          fileName: ocegress.cer
      caBundle:
        k8SecretName: occnp-gateway-secret
        k8NameSpace: occnp
        fileName: caroot.cer
      keyStorePassword:
```

```
    k8SecretName: occnp-gateway-secret
    k8NameSpace: occnp
    fileName: key.txt
  trustStorePassword:
    k8SecretName: occnp-gateway-secret
    k8NameSpace: occnp
    fileName: trust.txt
```

Table 3-74 Configurable Parameters for HTTPS Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.enableOutgoingHttps	Enabling it for outgoing https request	No	false	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.egressGwCertReloadEnabled	Egress Gateway Certificates Reload Enabled	No	false	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.egressGwCertReloadPath	Egress Gateway Certificates Reloading path	No	/egress-gw/store/reload	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.privateKey.k8SecretName	Name of the Kubernetes Secret which contains the private key for Policy.	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.privateKey.k8NameSpace	Name of the Kubernetes Namespace where the Kubernetes Secret containing the private key for Policy can be found	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.privateKey.rsa.fileName	rsa private key file name	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.privateKey.ecdsa.fileName	ecdsa private key file name	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.certificate.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.certificate.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	

Table 3-74 (Cont.) Configurable Parameters for HTTPS Configurations in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.service.ssl.certificate.rsa.fileName	rsa private key file name	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.certificate.ecdsa.fileName	ecdsa private key file name	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.caBundle.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.caBundle.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.caBundle.fileName	private key file name	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.keyStorePassword.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.keyStorePassword.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.keyStorePassword.fileName	File name that has password for keyStore	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.trustStorePassword.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.trustStorePassword.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.trustStorePassword.fileName	File name that has password for trustStore	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	

Here is a sample HTTPS configurations in egress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
# ---- HTTPS Configuration - BEGIN ----

  #Enabling it for egress https requests
  enableOutgoingHttps: true

  egressGwCertReloadEnabled: true
  egressGwCertReloadPath: /egress-gw/store/reload

  service:
    ssl:
      privateKey:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        rsa:
          fileName: rsa_private_key_pkcs1.pem
        ecdsa:
          fileName: ssl_ecdsa_private_key.pem
      certificate:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        rsa:
          fileName: ocegress.cer
        ecdsa:
          fileName: ssl_ecdsa_certificate.crt
      caBundle:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        fileName: caroot.cer
      keyStorePassword:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        fileName: key.txt
      trustStorePassword:
        k8SecretName: ocpcf-gateway-secret
        k8NameSpace: ocpcf
        fileName: trust.txt
  # ---- HTTPS Configuration - END ----
```

3.26 SCP Configuration

This section describes the customizatons that you can make in `occnp_custom_values_24.3.0.yaml` files to support SCP integration including SBI routing.

Important

- Routes supporting the SBI-Routing configuration are updated in Egress Gateway only when its configuration details are provided correctly. Example: `PeerSetConfiguration`, `PeerConfiguration`, `sbiroutingerrorcriteriasets`, and `sbiroutingerroractionsets`. Routes not supporting the SBI-Routing configuration are updated only when they have valid route definition.

To configure SBI-Routing:

- Use `Peerconfiguration` to define the list of peers to which Egress Gateway can send request. This list contains peers that support HTTP/ HTTP-Proxy / HTTPS communication.
- Use `Peersetconfiguration` to logically group the peers into sets. Each set contains a list of peers that support HTTP and HTTPS communication modes.
- Use `sbiRoutingErrorCriteriaSets` to define 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.
- Use `sbiRoutingErrorActionSets` to define an array of actionset, where each depicts an ID, action to be performed (Currently on REROUTE action is supported) and blacklist configurations.
- Use `Priority` for each peer in the set. Depending on the priority, it selects the primary, secondary, or tertiary peers to route requests.
- Use `SbiRoutingWeightBasedEnabled` parameter for weight factor when the priorities of the Peers from DNS SRV or static FQDN's configured at alternate route service are the same. This weight factor decides which Peer to be picked when their priorities are same.

  SbiRoutingWeightBasedEnabled feature works only when SCP address is configured as a virtual FQDN in the PeerConfiguration section. Egress Gateway uses the virtual FQDN of the SCP instance to query the alternate route service to get the list of alternate FQDN's along with their priority/weight assigned to it.

Note

- Egress Gateway accepts route configuration updates only if SBI-Routing feature is configured correctly.
- If the peer contains a virtual host address, Egress Gateway resolves the virtual host address using DNS-SRV query. If a peer is defined based on virtual host, then peerset can contain only one such peer for `httpconfiguration` and `httpsconfiguration`. User should not configure more than one virtual host based on peer in a given peerset for a given HTTP / HTTPS configuration.
- In case of peers based on virtual host, Egress Gateway does not consider priority values configured rather it retrieves priority from DNS-SRV records.

The following flags determine whether the configuration for routes and sbiRouting needs to be picked up from Helm

```
routeConfigMode: HELM
```

Configurations for SBI Routing

To enable and configure SBI Routing, perform the following configurations

- For `sbiRoutingDefaultScheme` parameter, the default value is `http`. The value specified in this field is considered when `3gpp-sbi-target-apiroot` header is missing.
- Now, configure a list of peers and peer sets. Each peer must contain `id`, `host`, `port`, and `apiPrefix`. 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.

```
sbiRouting:
    # Default scheme applicable when 3gpp-sbi-target-apiroot header is missing
    sbiRoutingDefaultScheme: http



    peerConfiguration:
      - id: peer1
        host: scp1.test.com
        port: 80
        apiPrefix: "/"
      - id: peer2
        host: scp2.test.com
        port: 80
        apiPrefix: "/"
    peerSetConfiguration:
      - id: set0
        httpConfiguration:
          - priority: 1
            peerIdentifier: peer1
          - priority: 2
            peerIdentifier: peer2
        httpsConfiguration:
          - priority: 1
            peerIdentifier: peer1
          - priority: 2
            peerIdentifier: peer2
```

> **Note**
>
> If required, users can configure more SCP instances in a similar way.

Route-level Configuration

Each route must have configured filters. In case, the SBIRouting functionality is required without the reroutes, then configure `routes[0].metadata.sbiRoutingEnabled=true`, SbiRouting in `filterName1`, and set arguments without the `errorHandling` section.

If SbiRouting functionality is required with the reroute mechanism, and the `SbiRoutingWeightBasedEnabled` parameter is enabled, then configure `routes[0].metadata.sbiRoutingEnabled=true` and `routes[0].metadata.SbiRoutingWeightBasedEnabled=true`, SbiRouting in `filterName1`, and set arguments with the `errorHandling` section.

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`.

The `sbiRoutingErrorCriteriaSets` 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 .

The `sbiRoutingErrorActionSets` contains an array of actionset, where each depicts an ID, action to be performed (Currently on REROUTE action is supported) and blacklist configurations.

Following is the SBI routing configuration with the Reroute functionality:

> **Note**
>
> Ensure to configure `sbiRoutingErrorCriteriaSets` and `sbiRoutingErrorActionSets`.

If you have peers configured in HTTPS, but you want to select https peers only but the interaction should be on http, then, `httpstargetOnly` must be set to `true` and `httpruriOnly` must be set to `true`.

If you have peers configured in HTTPS, but you want to select https peers only and interaction should be on https, then `httpstargetOnly` must be set to `true` and `httpruriOnly` must be set to `false`.

If you have peers configured in HTTP, but you want to select http peers only and interaction should be on http, then `httpstargetOnly` must be set to `false` and `httpruriOnly` must be set to `false`.

```
- id: nrf_direct
#      uri: https://dummy.dontchange
#      path: /nnrf-disc/**
#      order: 4
#      metadata:

#        httpsTargetOnly: false
#        httpRuriOnly: false
#        sbiRoutingEnabled: false
#        sbiRoutingWeightBasedEnabled: false
#      filterName1:
#        name: SbiRouting
#        args:
#          peerSetIdentifier: set0
#          customPeerSelectorEnabled: false
#          errorHandling:
#            - errorCriteriaSet: scp_direct2_criteria_1
#              actionSet: scp_direct2_action_1
#              priority: 1
```

```
#            - errorCriteriaSet: scp_direct2_criteria_0
#              actionSet: scp_direct2_action_0
#              priority: 2
#    - id: scp_route
```

Enable Rerouting

The Reroute mechanism works only for the incoming requests to Egress Gateway that are bound for SBI-Routing. The SBI-Routing bound requests must be rerouted to other instances of SBI based on certain response error codes or exceptions.

> **Note**
>
> The above configuration is effective only when `sbiRoutingEnabled` is set to `true`.

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`.

> **Note**
>
> `errorcriteriaset` and `actionset` must be configured for reroute to work.

To enable reroute functionality with `SBIrouting` , add the following values in the Helm configuration file:

```
routesConfig:
 - id: scp_direct2
   uri: https://dummy.dontchange2
   path: /<Intended Path>/**
   order: 3
   metadata:
    httpsTargetOnly: false
    httpRuriOnly: false
    sbiRoutingEnabled: false
   filterName1:
     name: SbiRouting
     args:
       peerSetIdentifier: set0
       customPeerSelectorEnabled: false
       errorHandling:
        - errorCriteriaSet: scp_direct2_criteria_1
          actionSet: scp_direct2_action_1
          priority: 1
        - errorCriteriaSet: scp_direct2_criteria_0
          actionSet: scp_direct2_action_0
          priority: 2

sbiRoutingErrorCriteriaSets:
 - id: scp_direct2_criteria_0
   method:
   - GET
```

```
    - POST
    - PUT
    - DELETE
    - PATCH
    exceptions:
    - java.util.concurrent.TimeoutException
    - java.net.UnknownHostException
  - id: scp_direct2_criteria_1
     method:
       - GET
       - POST
       - PUT
       - DELETE
       - PATCH
     response:
       cause:
         ignoreCauseIfMissing: false
         path: ".cause"
         reason:
          - "cause-1"
          - "cause-2"
       statuses:
         - statusSeries: 4xx
           status:
             - 400
      headersMatchingScript: "headerCheck,server,via,.*(SEPP|UDR).*"


sbiRoutingErrorActionSets:
 - id: scp_direct2_action_0
   action: reroute
   attempts:2
   blackList:
    enabled: false
    duration: 60000

 - id: scp_direct2_action_1
   action: reroute
   attempts:3
   blackList:
    enabled: false
    duration: 60000
```

`errorcriteria` can also be configured only with the status code. Following is the sample:

```
sbiRoutingErrorCriteriaSets:
 - id: scp_direct2_criteria_1
    method:
      - GET
      - POST
      - PUT
      - DELETE
      - PATCH
    response:
      statuses:
```

```
    - statusSeries: 4xx
      status:
        - 400
        - 404
    - statusSeries: 5xx
      status:
        - 500
        - 503
```

The path has to be configured per route. If /** is provided as a path, then all traffic except NRF will be SBI-routed. If a traffic to particular NF has to be SBI-routed, then the permanent start string of the URI has to be configured as a prefix. Example: For CHF, path: /nchf-spendinglimitcontrol/**. Similarly, for UDR, path: /nudr-dr/**.

> **Note**
>
> *Path, Reason, and ignoreCauseIfMissing* parameters must not be empty when cause is configured in the errorcriteriaset. The reason parameter must contain at least one reason. The statusSeries must be configured with only one status code.
>
> When `errorcriteria` is configured only with the status code, statusSeries can have multiple error codes.
>
> When the configuration is not successful, oc_egressgateway_routing_invalid_config_detected metrics is pegged and SBI Routing feature is disabled for the route for which this criteria set is configured.

Handling Server and Via Header

This is an enhancement to the SBI routing functionality. An additional alternate routing rule is applied to the Egress Gateway when the header check is included in the configuration. This can be configured through `sbiroutingerrorcriteriaset` and corresponding action can be taken by configuring `sbierroractionsets`.

To configure SBI Routing with Reroute functionality, see *Enable Rerouting*.

To enable Server and Via Header handling, add `headersMatchingScript` under the response entity within `sbiRoutingErrorCriteriaSets`.

> **Note**
>
> `headersMatchingScript` is a configuration that accepts a single string with comma seperated tokens.

Sample `sbiRoutingErrorCriteriaSets` configuration:

```
sbiRoutingErrorCriteriaSets:
  - id: scp_direct2_criteria_1
    method:
      - GET
      - POST
      - PUT
      - DELETE
```

```
    - PATCH
  response:
    statuses:
      - statusSeries: 4xx
        status:
          - 400
          - 404
      - statusSeries: 5xx
        status:
          - 500
          - 503
    headersMatchingScript: "headerCheck,server,via,.*(SEPP|UDR).*"
```

The `headersMatchingScript` contains the following tokens:

- `headerCheck` - The Validation function name. It must be constant.
- `server`: Header name
- `Via` : Header Name
- `*(SEPP|UDR).*` : Regex expression against which the server or via header will be matched against.

This `headersMatchingScript` configuration gets satisfied if the response contains server or via header and the content of the header matches the regex configured. Fot the criteriaset to be matched, the response method, response status code, and `headersMatchingScript` configuration should be satisfied. The `actionset` is configured to blacklist the peer if the correspondng criteriaset is matched.

Sample `sbiRoutingErrorActionSets` configuration:

```
sbiRoutingErrorActionSets:
  - id: scp_direct2_action_0
    action: reroute
    attempts: 2
    blackList:
      enabled: true
      duration: 60000
```

Once the `sbiRoutingErrorCriteriaSets` is selected, map this actionset to the selected criteriaset in the **errorHandling** section. The corresponding FQDN or Host in the server header value is blacklisted for the duration mentioned in the blackList section within the `sbiRoutingErrorActionSets`.

> **Note**
>
> While configuring the `sbiRoutingErrorCriteriaSets` with server header checks (`headersMatchingScript`), ensure that criteriaset has the highest priority in the **errorHandling** section. And, while configuring criteriaset without the server header checks, ensure to keep the `blackList.enabled` as false. This is done for server header blacklisting when server header check is required.

3.27 Alternate Route Service Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` files to configure alternate route service.

These configurations are applicable only when alternate route service is enabled.

To configure alternate route service, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-75 Configurable Parameters for Alternate Route Service Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.isIpv6Enabled	Set the value to true for this parameter when NF is deployed in IPv6 cluster.	No	false		Added in Release 22.3.0	
alternate-route.staticVirtualFqdns[0].name	Name of the virtual FQDN/FQDN	Optional		CNCPolicy & PCF	Added in Release 1.8.0	
alternate-route.staticVirtualFqdns[0].alternateFqdns[0].target	Name of the alternate FQDN mapped to above virtual FQDN	Yes, if "staticVirtualFqdns[0].name" is defined		CNCPolicy & PCF	Added in Release 1.8.0	
alternate-route.staticVirtualFqdns[0].alternateFqdns[0].port	Port of the alternate FQDN	Yes, if "staticVirtualFqdns[0].name" is defined	-	CNCPolicy & PCF	Added in Release 1.8.0	
alternate-route.staticVirtualFqdns[0].alternateFqdns[0].priority	Priority of the alternate FQDN	Yes, if "staticVirtualFqdns[0].name" is defined		CNCPolicy & PCF	Added in Release 1.8.0	
alternate-route.dnsSrvEnabled	Flag to enable the DNS-SRV query to coreDNS Server.	No	true	CNCPolicy & PCF	Added in Release 1.8.0	

Table 3-75 (Cont.) Configurable Parameters for Alternate Route Service Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.dnsSrvFqdnSetting.enabled	Flag to enable the usage of custom pattern for the FQDN while triggering DNS-SRV query	No	true	CNCPolicy & PCF	Added in Release 1.8.0	If this flag is set to false, then default value: "_{scheme}._tcp.{fqdn}." will be used.
alternate-route.dnsSrvFqdnSetting.pattern	Pattern of the FQDN which will used to format the incoming FQDN and Scheme while triggering DNS-SRV query	Yes if "dnsSrvFqdnSetting.enabled" is set to true	"_{scheme}._tcp.{fqdn}."	CNCPolicy & PCF	Added in Release 1.8.0	
egress-gateway.dnsSrv.host	Host of DNS Alternate Route Service	Conditional (If DnsSrv integration is required.)	5000	CNCPolicy & PCF	Added in Release 1.8.0	
egress-gateway.dnsSrv.port	Port of DNS Alternate Route Service	Conditional (If DnsSrv integration is required.)	5000	CNCPolicy & PCF	Added in Release 1.8.0	
egress-gateway.dnsSrv.scheme	Scheme of request that need to be sent to alternate route service.	Conditional (If DnsSrv integration is required.)	http	CNCPolicy & PCF	Added in Release 1.8.0	
egress-gateway.dnsSrv.errorCodeOnDNSResolutionFailure	Configurable error code to be used incase of DNS resolution failure.	Conditional (If DnsSrv integration is required.)	425	CNCPolicy & PCF	Added in Release 1.8.0	

Table 3-75 (Cont.) Configurable Parameters for Alternate Route Service Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfmanagement.alternateRouteServiceEnabled	Flag to tell nrf-client services if alternate route service is deployed or not. This flag should be set to true when the global.alternateRouteServiceEnable parameter is set as true.	No	false	CNCPolicy & PCF	Added in Release 1.8.0	Applicable only if Alternate Route Service is enabled.
nrf-client-nfdiscovery.alternateRouteServiceEnabled	Flag to tell nrf-client services if alternate route service is deployed or not. This flag should be set to true when the global.alternateRouteServiceEnable parameter is set as true.	No	false	CNCPolicy & PCF	Added in Release 1.8.0	Applicable only if Alternate Route Service is enabled.
alternate-route.isIpv6Enabled	Set the value to true for this parameter when NF is deployed in IPv6 cluster.	No	false	CNCPolicy & PCF	Added in Release 1.14.0	Applicable only if Alternate Route Service is enabled.

Here is a sample configurations for DNS-SRV in `occnp_custom_values_24.3.0.yaml` file:

```
#Static virtual FQDN Config
  staticVirtualFqdns:
    - name: https://abc.test.com
      alternateFqdns:
        - target: abc.test.com
          port: 5060
          priority: 10
        - target: xyz.test.com
          port: 5060
          priority: 20
    - name: http://xyz.test.com
      alternateFqdns:
        - target: xyz.test.com
          port: 5060
          priority: 10
        - target: abc.test.com
```

```
          port: 5060
          priority: 20  #Flag to control if DNS-SRV queries are sent to
coreDNS or not
  dnsSrvEnabled: true
  #Below configuration is for customizing the format of FQDN which will used
while querying coreDNS for SRV Records
  dnsSrvFqdnSetting:
    enabled: true  #If this flag is disabled, then default value of
"_{scheme}._tcp.{fqdn}." will be used for Pattern
    pattern: "_{scheme}._tcp.{fqdn}."   #Ex: _http._tcp.service.example.org.



egress-gateway:
  dnsSrv:
    host: 10.75.225.67
    port: 32081
    scheme: http
    errorCodeOnDNSResolutionFailure: 425


#Enabled when deployed in Ipv6 cluster
  isIpv6Enabled: false
```

3.28 Logging Configuration

This section describes the customizatons that you should make in `occnp_custom_values_24.3.0.yaml` files to configure logging.

To configure logging in ingress-gateway, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-76 Configurable Parameters for Logging Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.log.level.root	Log level for root logs	No	WARN	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	Applicable only when ingress-gateway is enabled.
ingress-gateway.log.level.ingress	Log level for ingress logs	No	INFO	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	Applicable only when ingress-gateway is enabled.

Table 3-76 (Cont.) Configurable Parameters for Logging Configuration in Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.log.level.oauth	Log level for oauth logs	No	INFO	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	Applicable only when ingress-gateway is enabled.

Here is a sample configurations for logging in ingress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
ingress-gateway:

  log:
    level:
      root: WARN
      ingress: INFO
      oauth: INFO
```

Table 3-77 Configurable Parameters for Logging Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.log.level.root	Log level for root logs	No	WARN	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	Applicable only when egress-gateway is enabled.
egress-gateway.log.level.egress	Log level for egress logs	No	INFO	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	Applicable only when egress-gateway is enabled.

Table 3-77 (Cont.) Configurable Parameters for Logging Configuration in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.log.level.oauth	Log level for oauth logs	No	INFO	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	Applicable only when egress-gateway is enabled.

Here is a sample configurations for logging in egress-gateway in `occnp_custom_values_24.3.0.yaml` file:

```
egress-gateway:

  log:
    level:
      root: WARN
      egress: INFO
      oauth: INFO
```

To configure logging in Alternate Route service, you should configure the following configurable parameters in custom-value.yaml file:

Table 3-78 Configurable Parameters for Logging Configuration in Alternate Route Service

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.log.level.root	Log level for root logs	No	WARN	CNC Policy & PCF	Added in Release 1.8.0	Applicable only when alternate route service is enabled.

Table 3-78 (Cont.) Configurable Parameters for Logging Configuration in Alternate Route Service

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.log.level.altroute	Log level for alternate route logs	No	INFO	CNC Policy & PCF	Added in Release 1.8.0	Applicable only when alternate route service is enabled.

Here is a sample configurations for logging in `occnp_custom_values_24.3.0.yaml` file:

```
alternate-route:

  log:
    level:
      root: WARN
      altroute: INFO
```

Configurations for Debug Tool

At the global level, the `extraContainers` flag can be used to enable or disable injecting extra container, that is, Debug Tool. Users can set DISABLED (default value) or ENABLED values for this parameter.

> **Note**
>
> To enable and configure Debug Tool, pre-deployment configurations need to be performed. For more information, see the "Using Debug Tool" section in *Oracle Communications Cloud Native Core Converged Policy Troubleshooting Guide*.

The following is a snippet from the `occnp_custom_values_24.3.0.yaml` file:

```
 # Use 'extraContainers' attribute to control the usage of extra
container(DEBUG tool).
  # Allowed Values: DISABLED, ENABLED
  extraContainers: DISABLED
```

Configuring Size Limit for Subscriber Activity Logging Mapping Table

At the global level, the `subsActMappingTableEntrySize` flag can be used to configure the size limit for the mapping table used for Subscriber Activity Logging in CNC Policy and PCF deployment modes. The default value for this parameter is set to 20.

The following is a snippet from the occnp_custom_values_24.3.0.yaml file:

```
# Variable to specify the size of Subscriber Activity Logging Mapping Table
 subsActMappingTableEntrySize: 20
```

3.29 Common Configurations for Services

This section describes the configurable parameters that can be used to perform some common configurations applicable to different services while deploying Cloud Native Core Policy.

Common Reference Configurations

You can configure some common parameters that are used in multiple services by configuring commonRef section under global parameters section of the Custom Values YAML file. The parameter values can be set under commonRef and same value is used by all the services through the reference variable for the configuration.

The following section describes the commonRef parameters for common configuration:

Table 3-79 Common Reference Configurations

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Notes
&configServerDB	Specifies the name of the config server database.	Yes	occnp_config_server	CNC Policy and PCF	
&commonConfigDB	Specifies the name of the common config database.	Yes	occnp_commonconfig	CNC Policy and PCF	
&commonCfgSvc.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	true	CNC Policy and PCF	
commonCfgSvc.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	8000	CNC Policy and PCF	Same value as global.servicePorts.cmServiceHttp.
&dbCommonConfig.dbHost	Specifies the MySQL database host for services.	Yes		CNC Policy and PCF	Same value as global.envMysqlHost.
&dbCommonConfig.dbPort	Specifies MySQL database port for services.	Yes		CNC Policy and PCF	Same value as global.envMysqlPort.
&dbCommonConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	occnp_commonconfig	CNC Policy and PCF	Same value as global.common.Ref.commonConfigDB

Table 3-79 (Cont.) Common Reference Configurations

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Notes
&dbCommonConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the <dbConfig.secretName>.	Yes	mysql-username	CNC Policy and PCF	
&dbCommonConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the <dbConfig.secretName>.	Yes	mysql-password	CNC Policy and PCF	

Common Configuration Service and Database configurations in Bulwark

The following section describes the customizable parameters for Common Configuration service in Bulwark:

Table 3-80 Common Configuration Service and Database configurations in Bulwark

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Notes
bulwark.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy and PCF	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
bulwark.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy and PCF	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-80 (Cont.) Common Configuration Service and Database configurations in Bulwark

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Notes
bulwark.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy and PCF	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
bulwark.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy and PCF	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
bulwark.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-privileged-db-pass	CNC Policy and PCF	Same value as `global.privilegedDbCredSecretName`
bulwark.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy and PCF	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
bulwark.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy and PCF	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
bulwark.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy and PCF	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Common Configuration Service and Database configurations in nrf-client-nfdiscovery

Table 3-81 Common Configuration Service and Database configurations in nrf-client-nfdiscovery

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfdiscovery.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfdiscovery.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfdiscovery.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfdiscovery.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-81 (Cont.) Common Configuration Service and Database configurations in nrf-client-nfdiscovery

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfdiscovery.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-db-pass	CNC Policy & PCF	Added in Release 1.11.0	Same value as `global.dbCredSecretName`
nrf-client-nfdiscovery.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfdiscovery.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfdiscovery.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Common Configuration Service and Database configurations in nrf-client-nfmanagement

Table 3-82 Common Configuration Service and Database configurations in nrf-client-nfmanagement

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfmanagement.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfmanagement.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfmanagement.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfmanagement.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-82 (Cont.) Common Configuration Service and Database configurations in nrf-client-nfmanagement

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfmanagement.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-privileged-db-pass	CNC Policy & PCF	Added in Release 1.11.0	Same value as `global.privilegedDbCredSecretName`
nrf-client-nfmanagement.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
nrf-client-nfmanagement.dbConfig.leaderPodDbName	Specifies the database name for LeaderPodDb database. This database is unique per site.	Yes (if multipod is supported for NRF client)	occnp_leaderPodDb	CNC Policy & PCF	Added in Release 22.2.0	
nrf-client-nfmanagement.dbConfig.networkDbName	Specifies the network database name.	Yes (if multipod is supported for NRF client)	occnp_release	CNC Policy & PCF	Added in Release 22.2.0	Same value as `global.releaseDbName`
nrf-client-nfmanagement.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-82 (Cont.) Common Configuration Service and Database configurations in nrf-client-nfmanagement

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfmanagement.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-82 (Cont.) Common Configuration Service and Database configurations in nrf-client-nfmanagement

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
nrf-client-nfmanagement.enablePDBSupport	To enable the multi-pod support for the nrf-client the enablePDBSupport should be set true	No	False	CNC Policy & PCF	Added in Release 22.4.x	Horizontal Pod Autoscaler (HPA) resource has been included to NfManagement with minReplicas and maxReplicas set as 2 by default. For this resource there are two scenarios: • Flag enableDBSupport enabled- This is multi-pod scenario and sets to minReplicas and maxReplicas for any value defined in values.yaml file. Currently, it is set as 2 for both properties by default. • Flag enablePDBSupport disabled- This is single-pod

Table 3-82 (Cont.) Common Configuration Service and Database configurations in nrf-client-nfmanagement

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
						scenario and set for both minReplicas and maxReplicas as 1.

Common Configuration Service and Database configurations in appinfo

Table 3-83 Common Configuration Service and Database configurations in appinfo

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
appinfo.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
appinfo.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-83 (Cont.) Common Configuration Service and Database configurations in appinfo

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
appinfo.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
appinfo.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
appinfo.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-db-pass	CNC Policy & PCF	Added in Release 1.11.0	Same value as `global.dbCredSecretName`
appinfo.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-83 (Cont.) Common Configuration Service and Database configurations in appinfo

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
appinfo.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the <dbConfig.secretName>.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
appinfo.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the <dbConfig.secretName>.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Non real-time based status API from the monitor service is dependent on the Prometheus. If Promethus-server and prometheus-kube-state-metrics is not working or installed properly then the non real-time API provides the wrong value.

It is recommended to use real-time DBstatus URIs because these URIs always provide the right values.

For example:

```
db_status_uri : http://occndbtier-db-monitor-svc:8080/db-tier/status/cluster/local/
realtime
realtime_db_status_uri : http://occndbtier-db-monitor-svc:8080/db-tier/status/cluster/
local/realtime
replication_status_uri : http://occndbtier-db-monitor-svc:8080/db-tier/status/
replication/realtime
```

Common Configuration Service and Database configurations in perf-info

Table 3-84 Common Configuration Service and Database configurations in perf-info

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
perf-info.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
perf-info.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
perf-info.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
perf-info.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-84 (Cont.) Common Configuration Service and Database configurations in perf-info

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
perf-info.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-db-pass	CNC Policy & PCF	Added in Release 1.11.0	Same value as `global.dbCredSecretName`
perf-info.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
perf-info.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
perf-info.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Common Configuration Service and Database configurations in ingress-gateway

Table 3-85 Common Configuration Service and Database configurations in ingress-gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
ingress-gateway.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
ingress-gateway.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
ingress-gateway.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-85 (Cont.) Common Configuration Service and Database configurations in ingress-gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-db-pass	CNC Policy & PCF	Added in Release 1.11.0	Same value as `global.dbCredSecretName`
ingress-gateway.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
ingress-gateway.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
ingress-gateway.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Common Configuration Service and Database configurations in egress-gateway

Table 3-86 Common Configuration Service and Database configurations in egress-gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
egress-gateway.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
egress-gateway.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
egress-gateway.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-86 (Cont.) Common Configuration Service and Database configurations in egress-gateway

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
egress-gateway.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-db-pass	CNC Policy & PCF	Added in Release 1.11.0	Same value as `global.dbCredSecretName`
egress-gateway.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
egress-gateway.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
egress-gateway.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Common Configuration Service and Database configurations in alternate-route

Table 3-87 Common Configuration Service and Database configurations in alternate-route

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.commonCfgClient.enabled	Specifies whether to enable or disable common config client for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
alternate-route.commonCfgServer.port	Specifies the common config server port for common config service.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
alternate-route.dbConfig.dbHost	Specifies the MySQL database host for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
alternate-route.dbConfig.dbPort	Specifies MySQL database port for services.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

Table 3-87 (Cont.) Common Configuration Service and Database configurations in alternate-route

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
alternate-route.dbConfig.secretName	Specifies kubernetes secret object name from which MYSQL username and password is picked.	Yes	occnp-db-pass	CNC Policy & PCF	Added in Release 1.11.0	Same value as `global.dbCredSecretName`
alternate-route.dbConfig.dbName	Specifies common config database name for services to store common configurations.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
alternate-route.dbConfig.dbUNameLiteral	Specifies the database literal name for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.
alternate-route.dbConfig.dbPwdLiteral	Specifies the database literal password for services to be used as per the `<dbConfig.secretName>`.	Yes	Same as the value provided in the Table 3-79	CNC Policy & PCF	Added in Release 1.11.0	To use a different values than the default value, remove the comment (#) from the respective parameters and edit the values.

> **Note**
>
> You can add additional parameters under the `dbConfig` for each service by adding key value pair after the `<<: *dbCommonConfig` text.
> The following snippet shows an example:
>
> ```
> dbConfig:
>  <<: *dbCommonConfig
>  <key>:<value>
> ```
>
> where, *<key>* is the parameter to be configured and *<value>* is the configured value for *<key>*.

3.30 Configuration for metrics

Global Metrics Configurations

Starting with CNE 1.9.0, if the user wants to enable monitoring via Prometheus, the following parameters must be configured:

Table 3-88 Global Configurations for Metrics

Parameter	Description	Notes
cncMetricsName	This parameter specifies the port, that is, `cnc-metrics` that Prometheus will scrape on.	This parameter is applicable to Converged, PCF, and PCRF deployment modes.
exposeObservabilityAtService	This parameter specifies whether to enable or disable Prometheus monitoring of services. By default. the value is set to false and services are not captured in Prometheus GUI.	This parameter is applicable to Converged, PCF, and PCRF deployment modes.

You can add prefix and suffix to metrics for CNC Policy services by using the following parameters:

```
metricPrefix: &metricPrefix 'occnp'
  metricSuffix: &metricSuffix ''
```

Table 3-89 Prefix and Suffix for Metrics

Parameter	Description	Notes
metricPrefix	This parameter specifies the prefix that you want to add to the metrics for CNC Policy services. **Default value**: occnp	This parameter is applicable to Converged, PCF, and PCRF deployment modes.
metricSuffix	This parameter specifies the suffix that you want to add to the metrics for CNC Policy services. **Default value**: empty string	This parameter is applicable to Converged, PCF, and PCRF deployment modes.

A reference is made to the metricPrefix and metricSuffix parameters, defined in the global section, under `nrf-client-nfdiscovery` and `nrf-client-nfmanagement` configurations.

> **Note**
>
> - If you choose to customize prefix, then it is required to align the NF delivered Grafana charts and Prometheus alerts with the updated metric names.
> - When you define a suffix for metrics, it may happen that the suffix appears in the middle of the metric name, and not towards the end. This is due to the fact that Micrometer library autogenerates some metrics and adds a suffix after the user-defined suffix.
>
> **Example**: If you define suffix as `occnp`, then the resulting metric name would appear in the system as `http_in_conn_response_occnp_total`.

3.31 Custom Container Name

This section describes how to customize the name of containers of a pod with a prefix and suffix. To do so, add the prefix and suffix to the k8sResource under global section of `occnp_custom_values_24.3.0.yaml` file:

```
global:
  k8sResource:
    container:
      prefix: ABCD
      suffix: XYZ
```

Then, after installing CNC policy, you will see the container names as shown below:

```
Containers:
  abcd-am-service-xyz:
```

3.32 Overload Manager Configurations

This section describes the customizatons that can be done in `occnp_custom_values_24.3.0.yaml` files to configure Overload Manager feature under perf-info.

Table 3-90 Configurable Parameters for overload Manager Configuration in Perf-Info

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
perf-info.overloadManager.enabled	Specifies whether to enable or disable overload reporting.	Optional	false	CNC Policy and PCF	Added in 1.12.1

Table 3-90 (Cont.) Configurable Parameters for overload Manager Configuration in Perf-Info

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
perf-info.envMysqlDatabase	Specifies the name of the database used for overload management. For georedundant setup, the value for this parameter must be unique for each site.	Conditional **Note:** This parameter value is required if the overload manager functionality is enabled by setting the value of `perf-info.overloadManager.enabled` to `true`.		CNC Policy and PCF	Added in 1.14.0
perf-info.overloadManager.ingressGatewaySvcName	Specifies the names of backend services	Conditional	occnp-ingress-gateway	CNC Policy and PCF	Added in 1.12.1
perf-info.overloadManager.ingressGatewayPort	Specifies the port number of Ingress Gateway	Mandatory	*svcIngressGatewayHttp	CNC Policy and PCF	Added in 1.12.1
perf-info.overloadManager.nfType	Specifies the NF type that is used to query configurration from common configuration server.	Mandatory	PCF	CNC Policy and PCF	Added in 1.12.1
perf-info.overloadManager.diamGatewayPort	Specifies the HTTP signaling port of Diameter Gateway, which is used for implementing overload control for Diameter interface.	Mandatory	*svcDiamGatewayHttp	CNC Policy, PCF, and PCRF	Added in 22.1.0

Here is a sample overloadManager configurations in perf-info in `occnp_custom_values_24.3.0.yaml` file:

```
perf-info:
  configmapPerformance:
    prometheus: ''
  # envMysqlDatabase is used for overload management.
  # If the customer does not use the overload management feature, this can be
```

```
ignored.
  envMysqlDatabase: ''
  overloadManager:
    enabled: false
    ingressGatewaySvcName: occnp-ingress-gateway
    ingressGatewayPort: *svcIngressGatewayHttp
    # nfType is used to query configuration from common cfg server
    nfType: PCF
    # diam Gateway overload management feature configurations
    diamGWPort: *svcDiamGatewayHttp
```

3.33 Detection and Handling Late Arrival Requests Configuration

This section describes the parameters that user can configure for detection and handling of late arrival requests.

You need to configure the following global and route level Helm parameters at AM and UE services:

Table 3-91 Configurable Parameters for SBI Timer Handling at AM and UE services

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
SBI_TIMER_ENABLED	Specifies whether the AM or UE service can generate the 3gpp-sbi headers related to the timer handling, if they are not received in the request.	Optional	false	CNC Policy & PCF	Added in Release 23.1.0

Table 3-92 Configurable Parameters for Late Arrival Handling at Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.isSbiTimerEnabled	Specifies whether to enable or disable SBI timer header enhancement. If the value of this parameter is set to true, SBI headers (3gpp-Sbi-Sender-Timestamp, 3gpp-Sbi-Max-Rsp-Time, and 3gpp-Sbi-Origination-Timestamp) are used along with route level (if configured) and global level request timeout to calculate final request timeout. After calculating the final request timeout, original values of 3gpp-Sbi-Sender-Timestamp, 3gpp-Sbi-Max-Rsp-Time and 3gpp-Sbi-Origination-Timestamp are published in custom headers Orig-3gpp-Sbi-Sender-Timestamp, Orig-3gpp-Sbi-Max-Rsp-Time and Orig-3gpp-Sbi-Origination-Timestamp respectively. If the value for this parameter is set to false, SBI headers are not taken into consideration even if they are present and no custom headers are published.	Optional	false	CNC Policy & PCF	Added in Release 1.15.0
ingress-gateway.publishHeaders	Specifies if the originating headers shall be populated and sent to the backend.	Optional	false	CNC Policy & PCF	Added in Release 1.15.0
ingress-gateway.sbiTimerTimezone	Specifies the time zone. It can be either set to GMT or ANY. If it is set to GMT then, the GMT should be specified in the header. If it is not specified, the time zone is assumed as GMT. If it is set to ANY then, the required time zone must be specified in the header. The timeout calculation is made as per the time zone specified in the header. If time zone is not specified then, the request is rejected and a gauge metric is pegged.	Optional	GMT	CNC Policy, PCF, & PCRF	Added in Release 1.15.0

The following is a snippet from the `occnp-1.15.0-custom-values.yaml` file:

```
isSbiTimerEnabled: false
publishHeaders: false
sbiTimerTimezone: GMT
routesConfig:
- id: demo
uri: https://demoapp.ocegress:8440/
path: /**
order: 1
#Below field is used to provide an option to enable/disable route
level xfccHeaderValidation, it will override global configuration for
xfccHeaderValidation.enabled
metadata:
# requestTimeout is used to set timeout at route level. Value
```

```
should be in milliseconds.
requestTimeout: 4000
# requiredTime is minimum time below which request will be
rejected if isSbiTimerEnabled is true. Value should be in milliseconds.
requiredTime: 3000
xfccHeaderValidation:
validationEnabled: false
oauthValidator:
enabled: false
svcName: "demo"
```

Table 3-93 Configurable Parameters for Late Arrival Handling at Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
egress-gateway.isSbiTimerEnabled	Specifies whether to enable or disable SBI timer header enhancement. If the value of this parameter is set to true, SBI headers (3gpp-Sbi-Sender-Timestamp, 3gpp-Sbi-Max-Rsp-Time, and 3gpp-Sbi-Origination-Timestamp) are used along with route level (if configured) and global level request timeout to calculate final request timeout. After calculating the final request timeout, original values of 3gpp-Sbi-Sender-Timestamp, 3gpp-Sbi-Max-Rsp-Time and 3gpp-Sbi-Origination-Timestamp are published in custom headers Orig-3gpp-Sbi-Sender-Timestamp, Orig-3gpp-Sbi-Max-Rsp-Time and Orig-3gpp-Sbi-Origination-Timestamp respectively. If the value for this parameter is set to false, SBI headers are not taken into consideration even if they are present and no custom headers are published.	Optional	false	CNC Policy & PCF	Added in Release 1.15.0
egress-gateway.sbiTimerTimezone	Specifies the time zone. It can be either set to `GMT` or `ANY`. If it is set to GMT then, the GMT should be specified in the header. If it is not specified, the time zone is assumed as GMT. If it is set to ANY then, the required time zone must be specified in the header. The timeout calculation is made as per the time zone specified in the header. If time zone is not specified then, the request is rejected and a gauge metric is pegged.	Optional	GMT	CNC Policy & PCF	Added in Release 1.15.0

To create Custom-Sbi-Sender-Timestamp it is necessary to add the following configuration to PCF ingress-gateway:

```
routesConfig:
    - id: sm_create_session_route
      uri: http://{{ .Release.Name }}-occnp-pcf-sm:
{{ .Values.global.servicePorts.pcfSmServiceHttp }}
```

```
      path: /npcf-smpolicycontrol/*/sm-policies
      order: 1
      method: POST
      readBodyForLog: true
      filters:
        subLog: true,CREATE,SM
        customReqHeaderEntryFilter:
          headers:
            - methods:
              - POST
              headersList:
                - headerName: 3gpp-Sbi-Message-Priority
                  defaultVal: 24
                  source: incomingReq
                  sourceHeader: 3gpp-Sbi-Message-Priority
                  override: false
                - headerName: Custom-Sbi-Sender-Timestamp
                  defaultVal: func:currentTime(EEE, d MMM yyyy HH:mm:ss.SSS
z,gmt)
                  source: incomingReq
                  sourceHeader: 3gpp-Sbi-Sender-Timestamp
                  override: false
```

Egress Gateway can be configured to avoid the headers being propagated to other NFs by using the following Helm configuration:

```
routesConfig:
  - id: udr_route
    uri: http://{{ .Values.global.udr_url }}:
{{ .Values.global.servicePorts.udrServiceHttp }}
    path: /nudr-dr/**
    order: 1
    removeRequestHeader:
      - name: 3gpp-Sbi-Max-Rsp-Time
      - name: 3gpp-Sbi-Origination-Timestamp
      - name: 3gpp-Sbi-Sender-Timestamp


  - id: chf_route
    uri: http://{{ .Values.global.chf_url }}:
{{ .Values.global.servicePorts.chfServiceHttp }}
    path: /nchf-spendinglimitcontrol/**
    order: 2
    removeRequestHeader:
      - name: 3gpp-Sbi-Max-Rsp-Time
      - name: 3gpp-Sbi-Origination-Timestamp
      - name: 3gpp-Sbi-Sender-Timestamp
```

Internal Microservices Timer Configurations

SM Service

```
- name: USER_SERVICE_CONNECTOR_TIMEOUT
value: "6000"
```

```
- name: POLICY_SERVICE_CONNECTOR_TIMEOUT
  value: "3000"
- name: BINDING_SERVICE_CONNECTOR_TIMEOUT
  value: "3000"
- name: PA_SERVICE_CONNECTOR_TIMEOUT
  value: "3000"
- name: SM_SERVICE_CONNECTOR_TIMEOUT
  value: "3000"
- name: BSF_CONNECTOR_TIMEOUT
  value: "3000"
- name: AF_CONNECTOR_TIMEOUT
  value: "3000"
- name: SMF_CONNECTOR_TIMEOUT
  value: "3000"
- name: NWDAF_AGENT_SERVICE_CONNECTOR_TIMEOUT
  value: "3000"
- name: JETTY_REQUEST_TIMEOUT
  value: "5000"
```

AM and UE Service

```
- name: AMF_CONNECTOR_TIMEOUT
  value: "3000"
- name: POLICY_SERVICE_CONNECTOR_TIMEOUT
  value: "3000"
- name: USER_SERVICE_CONNECTOR_TIMEOUT
  value: "6000"
- name: BULWARK_SERVICE_CONNECTOR_TIMEOUT
  value: "3000"
- name: JETTY_REQUEST_TIMEOUT
  value: "5000"
```

3.34 Server Header at Ingress Gateway

This section describes the parameters that you can configure to enable support for server header at Ingress Gateway.

Table 3-94 Configurable Parameters for Server Header at Ingress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
ingress-gateway.serverHeaderConfigMode	Specifies the mode of operation for configuring server header configuration. Since CNC Policy supports only REST mode of configuration, the feature flag "serverheaderdetails" must be enabled using REST API only. For more information, see the section "Server Header Support on Ingress Gateway" in *Oracle Communications Cloud Native Core Policy REST Specification Guide*.	Optional	REST	CNC Policy & PCF	Added in Release 22.1.0.

The following is a snippet from the `occnp-22.1.0-custom-values.yaml` file:

```
  #We support ServerHeader Configuration Mode as REST, the feature flag for
"server" header will need to be enabled through Rest configuration.
  serverHeaderConfigMode: REST
```

3.35 Usage Monitoring Service Configuration

This section describes the configurable parameters that can be customized for Usage Monitoring service.

Table 3-95 Configurable Parameters for Usage Monitoring Service Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment
usage-mon.resources.limits.ephemeralStorage	Specifies the minimum limit of Ephemeral Storage.	Optional	2Gi	CNC Policy, PCF, and PCRF
usage-mon.resources.limits.cpu	Specifies the minimum limit of CPU usage for Usage Monitoring.	Optional	4	CNC Policy, PCF, and PCRF
usage-mon.resources.limits.memory	Specifies the minimum limit of memory usage for Usage Monitoring.	Optional	2Gi	CNC Policy, PCF, and PCRF
usage-mon.resources.requests.cpu	Specifies the required limit of CPU usage for Usage Monitoring.	Optional	1	CNC Policy, PCF, and PCRF
usage-mon.resources.requests.memory	Specifies the required limit of memory usage for Usage Monitoring.	Optional	1Gi	CNC Policy, PCF, and PCRF
usage-mon.minReplicas	Specifies the minimum replicas for Usage Monitoring service.	Optional	1	CNC Policy, PCF, and PCRF
usage-mon.maxReplicas	Specifies the maximum replicas for Usage Monitoring service.	Optional	1	CNC Policy, PCF, and PCRF
usage-mon.livenessProbe.timeoutSeconds	Specifies the timeout (in seconds) for Liveness Probe.	Optional	3	CNC Policy, PCF, and PCRF
usage-mon.livenessProbe.failureThreshold	Specifies the wait time before performing first liveness probe by Kubelet.	Optional	3	CNC Policy, PCF, and PCRF
usage-mon.readinessProbe.failureThreshold	When a pod starts and the probe fails, Kubernetes waits for the threshold time before giving up.	Optional	3	CNC Policy, PCF, and PCRF
usage-mon.readinessProbe.timeoutSeconds	Specifies the timeout (in seconds) for Readiness Probe.	Optional	3	CNC Policy, PCF, and PCRF

Here is a sample configuration in `occnp_custom_values_24.3.0.yaml` file:

```
usage-mon:
  envMysqlDatabase: occnp_usagemon
  resources:
    limits:
      ephemeralStorage: 2Gi
      cpu: 4
      memory: 2Gi
    requests:
      cpu: 1
      memory: 1Gi
  minReplicas: 2
  maxReplicas: 4
  livenessProbe:
    timeoutSeconds: 3
    failureThreshold: 3
  readinessProbe:
    failureThreshold: 3
    timeoutSeconds: 3
```

3.36 Ingress Gateway Readiness Probe Configuration

This section describes the readiness probe configurations in the Ingress Gateway.

Ingress Gateway uses the readiness logic provided by Kubernetes to determine if a pod can accept or reject the incoming requests.

This feature enhances the readiness logic to determine the status of the pod. You can configure the feature in CNC Policy only through Helm. Based on the configurations, further checks are performed to determine the health of the pod.

An in-memory cache is maintained to store the updated configuration. The cache is updated if a profile is modified, added, or deleted. Ingress gateway periodically makes a GET request to the URLs that are configured using a scheduler that runs in the background. If the GET request is successful, then other checks can take place. Otherwise, the pod is marked as unhealthy.

> **Note**
>
> If there are any pending requests waiting for the response and readiness state of pod changes from READY to NOT_READY, then these requests are not considered.

The following table describes the parameters for configuring Readiness Probe in Ingress Gateway:

Table 3-96 Configurable Parameters for Readiness Probe Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Notes
readinessConfigMode	Specifies the mode to configure Readiness Probe in Ingress Gateway.	Mandatory	HELM	CNC Policy & PCF	
readinessCheckEnabled	Specifies whether to enable or disable Readiness Probe in Ingress Gateway.	Mandatory	false	CNC Policy & PCF	
readinessIndicatorPollingInterval	Specifies the time (in milliseconds) at which the **Readiness Cache** updates the readiness status of Ingress Gateway performing the probe or setting the readiness state value to onExceptionUsePreviousState.	Mandatory	3000	CNC Policy & PCF	
readinessConfig.serviceProfiles.id	Specifies the ID of the profile.	Mandatory	Readiness-profile-DBStatus	CNC Policy & PCF	

Table 3-96 (Cont.) Configurable Parameters for Readiness Probe Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Notes
readinessConfig.serviceProfiles.url	Specifies the URL to which the Readiness Probe is sent out to retrieve a response, on the basis of which the state of the Ingress Gateway pod will be decided.	Mandatory	http://{{ template "service-name-app-info" . }}:{{ .Values.global.containerPorts.appInfoHttp }}/status/category/realtimedatabase	CNC Policy & PCF	In addition to the default value, you can use the following values: 1. FQDN/IP Address. 2. Any microservice to define dependency upon: `http://<Helm Release Name>-<CNPCF Service Name>:9000/actuator/health/readiness`

Table 3-96 (Cont.) Configurable Parameters for Readiness Probe Configuration

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Notes
readinessConfig.serviceProfiles.responseCode	Specifies the response code expected from the service. If the actual response code matches with the configured one then pod will be marked as healthy.	Mandatory	200	CNC Policy & PCF	
readinessConfig.serviceProfiles.responseBody	Specifies the response expected from the service. If the actual response matches with the configured one then pod will be marked as healthy.	Mandatory	Running	CNC Policy & PCF	
readinessConfig.serviceProfiles.onExceptionUsePreviousState	Specifies whether to use the previous state of Ingress Gateway. When this flag is set to true, response and responseCode checks are not made irrespective of the previous state of service on Ingress Gateway.	Mandatory	true	CNC Policy & PCF	
readinessConfig.serviceProfiles.initialState	Specifies the inital state to be specified. It can be either ACCEPTING_TRAFFIC (to accept all incoming requests) or REFUSING_TRAFFIC (to reject all incoming requests).	Mandatory	ACCEPTING_TRAFFIC	CNC Policy & PCF	
readinessConfig.serviceProfiles.requestTimeout	Specifies the timeout value of the probe in milliseconds.	Optional	2000	CNC Policy & PCF	

Check the following when the Ingress Gateway pod comes up:

1. If the service profiles are not configured, then the readiness probe of Ingress Gateway fails and the pod is marked as unhealthy.
2. If the service profiles are configured, check the mandatory parameters: `id`, `url`, `onExceptionUsePreviousState`, and `initialState` for thieir validity. If they are invalid, then the pod is marked as unhealthy.

> **Note**
>
> You must configure one of these parameters: `responseBody` or `responseCode` in the service profile. If any of these checks fail, then the pod does not come up in the case of Helm based configuration.

3. If there is any error like connection failure or connection timeout during making a request to backend service, then `onExceptionUsePreviousState` attribute is checked. If it is set to true, then previous state is used for that URL. If previous state is unavailable, then initial state is used. If `onExceptionUsePreviousState` is false, then the pod is marked as unhealthy.

3.37 Creating Custom Headers

This section provides information on how to create custom headers for routes in CNC Policy.

You can customize the headers present in the requests and responses based on the type of HTTP methods. This framework modifies the outgoing request or response by adding a new header either with a static value or with a value based on incoming request or response headers at entry or exit points.

By setting the `override` attribute value as true, you can override the existing headers. It is an optional attribute. It adds a new header or replaces the value of an existing header if one of the value is mapped to the source header. The value of this attribute is false by default.

The following is a sample configuration for custom header in `sm_delete_session_route`:

```
- id: sm_delete_session_route
      uri: http://{{ .Release.Name }}-occnp-pcf-sm:
{{ .Values.global.servicePorts.pcfSmServiceHttp }}
      path: /npcf-smpolicycontrol/*/sm-policies/{policy-id}/delete
      order: 2
      method: POST
      filters:
        subLog: true,DELETE,SM
        customReqHeaderEntryFilter:
          headers:
            - methods:
              - POST
              headersList:
                - headerName: 3gpp-Sbi-Message-Priority
                  defaultVal: 16
                  source: incomingReq
                  sourceHeader: 3gpp-Sbi-Message-Priority
                  override: false
```

> **Note**
>
> The attributes `headerName` and `sourceHeader` are case sensitive. Ensure that the value is same as in the incoming request or response in order to extract values from or override value of any particular header.

3.37.1 Custom Header Name for UDR Group Id

The following table lists the parameters to define customizied header name in the incoming requests for AM/UE/SM services create session routes.

Table 3-97 Routes Configurations

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment
routesConfig.id	Routes Configurations for Policy services.	M	SM service: sm_create_session_route AM service: am_create_session_route UE service: ue_create_session_route	CNC Policy & PCF
routesConfig.id.filters.customReqHeaderEntryFilter.headers.methods.headerList.headerName	Header name in the incoming requests.	M	oc-policy-udr-group-id-list	CNC Policy & PCF
routesConfig.id.filters.customReqHeaderEntryFilter.headers.methods.headerList.sourceHeader	Source header name in the incoming request.	M	oc-policy-udr-group-id-list	CNC Policy & PCF

An example of default header structure in the `occnp_custom_values_24.3.0.yaml` file:

```
routesConfig:
    - id: sm_create_session_route
      uri: http://{{ .Release.Name }}-occnp-pcf-sm:
{{ .Values.global.servicePorts.pcfSmServiceHttp }}
      path: /npcf-smpolicycontrol/*/sm-policies
      order: 1
      method: POST
      readBodyForLog: true
      filters:
        subLog: true,CREATE,SM
        customReqHeaderEntryFilter:
          headers:
            - methods:
              - POST
              headersList:
                - headerName: 3gpp-Sbi-Message-Priority
                  defaultVal: 24
                  source: incomingReq
                  sourceHeader: 3gpp-Sbi-Message-Priority
                  override: false
                - headerName: oc-policy-udr-group-id-list
                  source: incomingReq
```

```
                    sourceHeader: oc-policy-udr-group-id-list
                    override: false
```

3.38 Configurable Error Codes

This section describes the parameters that you can customize for configurable error codes.

Table 3-98 Configurable Parameters for Error Codes - Global

Parameter	Description	Mandatory/ Optional Parameter	Default Value
configurableErrorCodes.enabled	Specifies whether to enable or disable configurable error codes that can be used for messages over Ingress Gateway and Egress Gateway.	Optional	false

For a given error scenario, you can define exceptionType, errorCode, errorDescription, errorCause, and errorTitle as shown in the following snippet from the `occnp_custom_values_24.3.0.yaml` file.

Following is the configuration for error codes at global level:

```
ingress-gateway:

  configurableErrorCodes:
    enabled: true
    errorScenarios:
      - exceptionType: "XFCC_HEADER_INVALID"
        errorProfileName: "ERR_1300"
      - exceptionType: "XFCC_HEADER_VALIDATION_FAILURE"
        errorProfileName: "ERR_1300"

  errorCodeProfiles:
    - name: ERR_1300
      errorCode: 401
      errorCause: "xfcc header is invalid"
      errorTitle: "Invalid XFCC Header"
      errorDescription: "Invalid XFCC Header"
```

Following points must be noted for the global level configuration:

- To enable configurable error code global configurableErrorCodes flag must be set to true. If this flag is false then the hardcoded error codes will be returned when an exception is encountered at Ingress and Egress Gateways.
- If global configurableErrorCodes flag is set to true then atleast one entry must be configured in the errorScenarios section.
- For every Exception in errorScenarios there must be an error profile with that exceptionType. Moreover, a profile with that name must be configured in errorCodeProfiles section example - if errorProfileName: "ERR_1300" has been configured then a profile with name ERR_1300 must be present in errorCodeProfiles section.
- ExceptionType field in global and in the routes section is non configurable. These are hard coded values and can be taken from custom.yaml file.

Following is the configuration for error codes at route level:

```
routesConfig:
   - id: route1
     uri:
     path: /dummy/*/dummies
     order: 1
     method: POST
     metadata:
       configurableErrorCodes:
         enabled: true
         errorScenarios:
           - exceptionType: "XFCC_HEADER_INVALID"
             errorProfileName: "ERR_1300"
           - exceptionType: "XFCC_HEADER_VALIDATION_FAILURE"
             errorProfileName: "ERR_1300"
```

Following points must be noted for the route level configuration:

- If Route level is enabled, it has higher precedence over global level.
- For Route level configurable error codes to work, configurableErrorCodes flag must be set to true both at route level as well as global level.
- For a given exception at gateway, if there is no match at route level then global level is matched. If there is no match at global level, then hardcoded error values are returned.
- If configurableErrorCodes flag is disabled for a specific route and if an exception occurs at that route then hardcoded error responses will be returned irrespective of what is defined at global level.

> **Note**
>
> For every errorScenario, `exceptionType` and `errorCode` are manadatory parameter configurations.

Configurable Error Codes - SCP Integration

The following parameters are added under Egress Gateway for SCP related configurations. These error code configurations are included in error response from Egress Gateway when it is unable to resolve DNS successfully:

```
dnsSrv:
    port: *svcAlternateRouteServiceHttp
```

For more information about the error codes, see Configurable Error Codes.

3.39 Controlled Shutdown Configurations

This section describes the customizatons that can be done in `occnp_custom_values_24.3.0.yaml` files to configure controlled shutdown feature.

Table 3-99 Global Parameter for Controlled Shutdown

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment
global.enableControlledShutdown	Specifies whether to enable or disable the Controlled Shutdown feature.	Mandatory	False	CNC Policy & PCF

Table 3-100 Configurable Parameters for Controlled Shutdown in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment
egress-gateway.errorcodeprofiles	Error defined by the user	Optional	NA	CNC Policy & PCF
egress-gateway.errorcodeprofiles.name	Name of the error profile	Optional	NA	CNC Policy & PCF
egress-gateway.errorcodeprofiles.errorCode	Error code of the error profile	Optional	NA	CNC Policy & PCF
egress-gateway.errorcodeprofiles.errorCause	Cause of the error profile	Optional	NA	CNC Policy & PCF
egress-gateway.errorcodeprofiles.errorTitle	Title of the error profile	Optional	NA	CNC Policy & PCF
egress-gateway.errorcodeprofiles.retry-after	Retry-after value of the error profile	Optional	NA	CNC Policy & PCF
egress-gateway.errorcodeprofiles.errorDescription	Description of the error profile	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig	Routes configuration processed by the Egress Gateway	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig.id	ID of the route	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig.uri	URI of the route	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig.path	Path of the route	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig.order	Order in which the routes will be processed	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig.filters	Conditions on the routes	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig.filters.controlledShutdownFilter	Filter specified for Controlled Shutdown feature	Optional	NA	CNC Policy & PCF

Table 3-100 (Cont.) Configurable Parameters for Controlled Shutdown in Egress Gateway

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment
egress-gateway.routesConfig.filters.controlledShutdownFilter.applicableShutdownStates	States of Controlled shutdown feature, that is COMPLETE_SHUTDOWN	Optional	NA	CNC Policy & PCF
egress-gateway.routesConfig.filters.controlledShutdownFilter.unsupportedOperations	Operations which needs not be supported for controlled shutdown feature	Optional	NA	CNC Policy & PCF
egress-gateway.controlledShutdownErrorMapping	Array containing route ID and error profile name	Optional	NA	CNC Policy & PCF
egress-gateway.controlledShutdownErrorMapping.routeErrorProfileList	List of route ID and their corresponding error profile names	Optional	NA	CNC Policy & PCF
egress-gateway.controlledShutdownErrorMapping.routeErrorProfileList.routeId	Route ID on which the error profile name needs to be mapped	Optional	NA	CNC Policy & PCF
egress-gateway.controlledShutdownErrorMapping.routeErrorProfileList.errorProfileName	Error name from the error code profiles to be mapped in route ID	Optional	NA	CNC Policy & PCF

Here is a sample Error Codes configuratiom in Egress Gateway in the `occnp_custom_values_24.3.0.yaml` file:

```
errorcodeprofiles:
  - name: error300,
    errorCode: 300,
    errorCause: "",
    errorTitle: "",
    retry-after: "",
    errorDescription: ""
  - name: error500,
    errorCode: 500,
    errorCause: "",
    errorTitle: "",
    retryAfter: "",
    errorDescription: ""
```

Here is a sample routes configuration for Controlled Shutdown in Egress Gateway in the `occnp_custom_values_24.3.0.yaml` file:

```
routesConfig:
  - id: nrf_state
```

```
    uri: https://dummy.dontchange_1
    path: /nnrf-nfm/*
    order: 1
  - id: sampleRoute
    uri: https://dummy.dontchange_2
    path: /**
    order: 2
    metadata:
      httpsTargetOnly: false
      httpRuriOnly: false
      sbiRoutingEnabled: true
      oauthEnabled: false
    filterNameControlShutdown:
      name: ControlledShutdownFilter
      args:
        applicableShutdownStates:
          - COMPLETE_SHUTDOWN
        unsupportedOperations:
          - GET
          - PUT
          - PATCH
          - POST
          - DELETE
```

Here is a sample Error Codes Mapping configuratiom in Egress Gateway in the `occnp_custom_values_24.3.0.yaml` file:

```
controlledShutdownErrorMapping:
  routeErrorProfileList:
    - routeId: sampleRoute
      errorProfileName: "error503"
```

3.40 Perf-Info Configuration

Configurations for Perf-Info Capacity

This section provides information on how to configure the overall capacity and the capacity for individual services of perf-info in CNC Policy.

You can configure the perf-info capacity using the following parameters under the perf-info section of the `occnp_custom_values_24.3.0.yaml` file:

Table 3-101 Configurations for Perf-Info Capacity

Parameter	Description	Notes
perf-info.global.capacityConfig.overall	The overall capacity for the perf-info service.	If this value is not configured, then the default capacity value is considered.
perf-info.global.capacityConfig.serviceLevel	The service specific capacity for individual CNC Policy services.	If this value is not configured, then the default capacity value is considered.

Table 3-101 (Cont.) Configurations for Perf-Info Capacity

Parameter	Description	Notes
perf-info.global.capacityConfig.default	The default capacity.	The default capacity valuethat is used when the `overall` and `serviceLevel` values are not configured. **Default value:** 100 **Note:** If no value is set for the parameter then the default value used.

The following is a sample configuration for perf-info capacity configuration in `perf-info`:

```
capacityConfig:
          overall:100

serviceLevel:'{"occnp_pcf_am":100,"occnp_pcf_sm":100","pcf_ueservice":100}'

          default:100
```

CNE Configurations for Perf-Info

To configure label names, you should configure the following configurable parameters in `occnp_custom_values_24.3.0.yaml` file:

Table 3-102 Configurable Parameters for Logging Configuration in Prometheus

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
perf-info.tagNamespace	Specifies the Kubernetes namespace.	Mandatory	`kubernetes_namespace` (for CNE 1.8.0) `namespace` (for CNE 1.9)	CNC Policy, PCF	Added in 1.15.0
perf-info.tagContainerName	Specifies the tag used for specifying name of the container.	Mandatory	`container_name` (for CNE 1.8.0) `container` (for CNE 1.9)	CNC Policy, PCF	Added in 1.15.0

Table 3-102 (Cont.) Configurable Parameters for Logging Configuration in Prometheus

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
perf-info.tagServiceName	Specifies the tag used for specifying name of the service.	Mandatory	`kubernetes_name` (for CNE 1.8.0) `service` (for CNE 1.9)	CNC Policy, PCF	Added in 1.15.0

The following is a snippet from the `occnp_custom_values_24.3.0.yaml` file:

```
#Values for CNE 1.8 {tagNamespace: kubernetes_namespace, tagContainerName:
container_name, tagServiceName: kubernetes_name}
#Values for CNE 1.9 {tagNamespace: namespace, tagContainerName: container,
tagServiceName: service}
tagNamespace: kubernetes_namespace
tagContainerName: container_name
tagServiceName: kubernetes_name
```

3.41 Configurations for NodeSelector

Kubernetes `nodeSelector` feature is used for manual pod scheduling. A Policy pod is assigned to only those nodes that have label(s) identical to label(s) defined in the nodeSelector.

To list all the labels attached to a node you can run:

```
kubectl describe node pollux-k8s-node-1
Name:               pollux-k8s-node-1
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    kubernetes.io/hostname=pollux-k8s-node-1
                    kubernetes.io/os=linux
                    topology.kubernetes.io/region=RegionOne
                    topology.kubernetes.io/zone=nova
```

The default labels attached to kubernetes nodes are displayed. In order to assign a pod to the node in policy, you need to set custom configurations in `occnp_custom_values_24.3.0.yaml` file.

For all the Policy services you can configure the nodeselector either at global or local services section of the custom-values.yaml file. If global and local nodeselector configurations are performed in the custom-values.yaml file then the global configurations takes precedence and shall be considered by the Policy application.

Table 3-103 Global Configurations for Node Selector

Parameter	Description	Values	Notes

global.nodeSelection	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• ENABLED
		• DISABLED
		Default Value: DISABLED
			global:
			  nodeSelection: ENABLED
			  nodeSelector:
			  nodeKey: key
			  nodeValue: value
			
			For example:
			
			global:
			  nodeSelection: ENABLED
			  nodeSelector:
			  nodeKey: 'kubernetes.io/os'
			  nodeValue: 'linux'

global.nodeSelector.nodeKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
global.nodeSelector.nodeValue	Specifies valid value pair for the above key for a label for a particular node.	'Not Applicable	

Table 3-104 Local Configurations for Node Selector

Parameter	Description	Values	Notes

am-service.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• ENABLED
		• DISABLED
		Default Value: DISABLED
			am-service:
			  nodeSelectorEnabled: true
			  nodeSelectorKey: key
			  nodeSelectorValue: value
			
			For example:
			
			am-service:
			  nodeSelectorEnabl

am-service.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

am-service.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	`ed: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue : linux`

bulwark.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`bulwark:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue : linux`
			`nodeSelection: ENABLED`
			`nodeSelector:`
			`key: value`
			
			For example:
			
			`bulwark:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue : linux`
			`nodeSelection: ENABLED`
			`nodeSelector:`
			
			`'kubernetes.io/os': 'linux'`

bulwark.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
bulwark.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node	Not Applicable	
bulwark.nodeSelection	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values: • ENABLED • DISABLED **Default Value**: DISABLED	
bulwark.nodeSelector	Specifies the key value pair for a label of a particular node.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes
sm-service.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values: • true • false **Default Value**: false	sm-service: nodeSelectorEnabled: true nodeSelectorKey: key nodeSelectorValue: value For example: sm-service: nodeSelectorEnabled: true nodeSelectorKey: kubernetes.io/os nodeSelectorValue: linux
sm-service.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
sm-service.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	
ue-service.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values: • true • false **Default Value**: false	ue-service: nodeSelectorEnabled: true nodeSelectorKey: key nodeSelectorValue: value Sample Configuration: ue-service: nodeSelectorEnabled: true
ue-service.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

ue-service.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue : linux`

user-service.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`user-service:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: key`
			
			`nodeSelectorValue : value`
			
			For example:
			
			`user-service:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue : linux`

user-service.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
user-service.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

config-server.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`config-server:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: key`
			
			`nodeSelectorValue : value`

config-server.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

config-server.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	For example:
			
			`config-server:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue: linux`

queryservice.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`queryservice:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: key`
			
			`nodeSelectorValue: value`
			
			For example:
			
			`queryservice:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue: linux`

queryservice.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
queryservice.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

cm-service.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`cm-service:`
			
			`nodeSelectorEnabled: true`

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes
cm-service.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	`nodeSelectorKey: key`

cm-service.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	`nodeSelectorValue: value`
			
			For example:
			
			`cm-service:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: kubernetes.io/os`
			`nodeSelectorValue: linux`

audit-service.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`audit-service:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: key`
			`nodeSelectorValue: value`
			
			For example:
			
			`audit-service:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: kubernetes.io/os`
			`nodeSelectorValue: linux`

audit-service.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
audit-service.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

nrf-client.nrf-client-nfdiscovery.global.deploymentNrfClientService.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			nrf-client:
			  nrf-client-nfdiscovery:
			    global:
			
			ephemeralStorageLimit: 1024
			
			deploymentNrfClientService:
			
			nodeSelectorEnabled: true
			
			nodeSelectorKey: key
			
			nodeSelectorValue: value
			
			For example:
			
			nrf-client:
			  nrf-client-nfdiscovery:
			    global:
			
			ephemeralStorageLimit: 1024
			
			deploymentNrfClientService:
			
			nodeSelectorEnabled: true
			
			nodeSelectorKey: kubernetes.io/os
			
			nodeSelectorValue: linux

nrf-client.nrf-client-nfdiscovery.global.deploymentNrfClientService.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
nrf-client.nrf-client-nfdiscovery.global.deploymentNrfClientService.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	
nrf-client.nrf-client-nfmanagement.global.deploymentNrfClientService.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values: • true • false **Default Value**: false	nrf-client:

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

nrf-client.nrf-client-nfmanagement.global.deploymentNrfClientService.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	`nrf-client-nfmanagement:`
			`global:`
			
			`deploymentNrfClientService:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: key`
			
			`nodeSelectorValue: value`
			
			For example:
			
			`nrf-client:`
			`nrf-client-nfmanagement:`
			`global:`
			
			`deploymentNrfClientService:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue: linux`

nrf-clientnrf-client-nfmanagement.global.deploymentNrfClientService.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

appinfo.nodeSelection	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• ENABLED
		• DISABLED
		Default Value: DISABLED
			`appinfo:`
			`nodeSelection: ENABLED`
			`nodeSelector:`
			`key: value`

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

appinfo.nodeSelector	Specifies the key value pair for a label of a particular node.	Not Applicable	For example:
			
			`appinfo:`
			`  nodeSelection:`
			`ENABLED`
			`  nodeSelector:`
			
			`'kubernetes.io/`
			`os': 'linux'`

perf-info.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`perf-info:`
			
			`nodeSelectorEnabl`
			`ed: true`
			
			`nodeSelectorKey:`
			`key`
			
			`nodeSelectorValue`
			`: value`
			
			For example:
			
			`perf-info:`
			
			`nodeSelectorEnabl`
			`ed: true`
			
			`nodeSelectorKey:`
			`kubernetes.io/os`
			
			`nodeSelectorValue`
			`: linux`

perf-info.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
perf-info.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

diam-connector.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`diam-connector:`
			
			`nodeSelectorEnabl`
			`ed: true`
			
			`nodeSelectorKey:`
			`key`

diam-connector.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

diam-connector.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	`nodeSelectorValue: value`
			
			For example:
			
			`diam-connector:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: kubernetes.io/os`
			`nodeSelectorValue: linux`

diam-gateway.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`diam-gateway:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: key`
			`nodeSelectorValue: value`
			
			For example:
			
			`diam-gateway:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: kubernetes.io/os`
			`nodeSelectorValue: linux`

diam-gateway.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
diam-gateway.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes
policyds.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values: • true • false **Default Value**: false	policyds: nodeSelectorEnabled: true nodeSelectorKey: key nodeSelectorValue: value For example: policyds: nodeSelectorEnabled: true nodeSelectorKey: kubernetes.io/os nodeSelectorValue: linux
policyds.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
policyds.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	
ldap-gateway.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values: • true • false **Default Value**: false	ldap-gateway: nodeSelectorEnabled: true nodeSelectorKey: key nodeSelectorValue: value For example: ldap-gateway: nodeSelectorEnabled: true
ldap-gateway.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

ldap-gateway.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	'Not Applicable	`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue : linux`

pre-service.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`pre-service:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: key`
			
			`nodeSelectorValue : value`
			
			For example:
			
			`pre-service:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue : linux`

pre-service.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
pre-service.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

pcrf-core.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`pcrf-core:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: key`
			
			`nodeSelectorValue : value`

pcrf-core.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes

pcrf-core.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	For example:
			
			`pcrf-core:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue: linux`

soap-connector.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`soap-connector:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: key`
			
			`nodeSelectorValue: value`
			
			For example:
			
			`soap-connector:`
			
			`nodeSelectorEnabled: true`
			
			`nodeSelectorKey: kubernetes.io/os`
			
			`nodeSelectorValue: linux`

soap-connector.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
soap-connector.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

binding.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`binding:`
			
			`nodeSelectorEnabled: true`

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes
binding.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	`nodeSelectorKey: key`

binding.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	`nodeSelectorValue : value`
			
			For example:
			
			`binding:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: kubernetes.io/os`
			`nodeSelectorValue : linux`

notifier.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values:
		• true
		• false
		Default Value: false
			`notifier:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: key`
			`nodeSelectorValue : value`
			
			For example:
			
			`notifier:`
			`nodeSelectorEnabled: true`
			`nodeSelectorKey: kubernetes.io/os`
			`nodeSelectorValue : linux`

notifier.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
notifier.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

Table 3-104 (Cont.) Local Configurations for Node Selector

Parameter	Description	Values	Notes
usage-mon.nodeSelectorEnabled	Specifies if pods needs to assigned to a specific node manually or not.	Allowed Values: • true • false **Default Value**: false	usage-mon: nodeSelectorEnabled: true nodeSelectorKey: key nodeSelectorValue: value For example: usage-mon: nodeSelectorEnabled: true nodeSelectorKey: kubernetes.io/os nodeSelectorValue: linux
usage-mon.nodeSelectorKey	Specifies a valid key that is a node label of a particular node in the cluster.	Not Applicable	
usage-mon.nodeSelectorValue	Specifies valid value pair for the above key for a label of a particular node.	Not Applicable	

3.42 Configurations for Anti-Affinity Rule

This section describes the configuration parameters required for pod anti-affinity scheduling. These are configurable parameters in the custom-values.yaml file.

Table 3-105 Configurable Parameters for Pods Anti-Affinity

Parameter	Description	Mandatory Parameter(Y/N)	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
preferredDuringSchedulingIgnoredDuringExecution	Specifies that the scheduler tries to find a node that meets the anti-affinity rule	N		CNC Policy	Added in Release 22.3.0	If a matching node is not available, the scheduler still schedules the Pod.

Table 3-105 (Cont.) Configurable Parameters for Pods Anti-Affinity

Parameter	Description	Mandatory Parameter(Y/N)	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
weight	For each instance of the preferredDuringSchedulingIgnoredDuringExecutionaffinity type, you can specify a weight between 1 and 100	N	100	CNC Policy	Added in Release 22.3.0	
matchExpressions.key	Defines the rules for constraining a Pod. The scheduler avoids schedulingPods having configured key.	N	NA	CNC Policy	Added in Release 22.3.0	
matchExpressions.values	The scheduler avoids schedulingPods having configured value.	N	NA	CNC Policy	Added in Release 22.3.0	
topologyKey	The key for the node label used to specify the domain	N	NA	CNC Policy	Added in Release 22.3.0	

Sample Affinity Rule:

```
affinity:
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        podAffinityTerm:
          labelSelector:
            matchExpressions:
            - key: "app.kubernetes.io/name"
              operator: In
              values:
                - {{ template "chart.fullname" .}}
            topologyKey: "kubernetes.io/hostname
```

3.43 Configuration Parameters for IPv6

Table 3-106 Configurable Parameters for IPv6

Parameter	Description	Mandatory Parameter	Default Value	Value to Enable IPv6	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.isIpvSixSetup	Enable HTTP communication in IPv6	No	false	True	CNC Policy, PCF, & PCRF	Added in Release 23.2.x	This value must be set to "true" if you are going to require HTTP communication over IPv6.
diam-gateway.envSupportedIpAddressType	Distinguish between the IP address types for which diam-gw would enable connectivity and not depend on the IP address type of the infrastructure.	No	IPv4	IPv6	CNC Policy, PCF, & PCRF	Added in Release 22.1.0	This parameter must be set to IPv6 if the diam-gw connectivity will be exclusively in "IPv6" or "BOTH" if the connectivity will be for IPv4 and IPv6.

Note

You must enable the IPv6 related parameters in Alternate Route, Ingress Gateway, and Egress Gateway services configurations.

Note

When Policy is being installed in a dual stack environment with IPv6 enabled, it is necessary to edit each service by changing "ipFamilies" and "ipFamilyPolicy" as follows:

```
ipFamilies:
 - IPv6
 - IPv4
ipFamilyPolicy: RequireDualStack
```

3.44 Bulwark Service Configuration

This section describes the configuration parameters for Bulwark service.

Table 3-107 Configurable Parameters for Bulwark Service

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment
polling.interval	This parameter is used to configure the time interval between which Bulwark service polls to know if an existing lock is released. **Default value**: 20 ms **Recommended range**: 1-100 ms That is, by default Bulwark service polls for the lock every 20 ms. This interval is recommended to be configured to any value between 1 and 100 ms.	Optional	20 ms	CNC Policy, PCF, and PCRF
polling.maxLockAttempts	This parameter is used to configure the maximum number of times Bulwark service can retry to acquire the lock when a request for lock acquisition fails. **Default value**: 5	Optional	5	CNC Policy, PCF, and PCRF
bulwark.congestion.responseCode	When Bulwark pod is in congestion state, the response code for the rejected requests can be configured using this parameter. The user can configure this parameter with other supported 5xx response codes.	Optional	503	CNC Policy, PCF, and PCRF

3.45 Configurations Parameters for Undertow Server Queue

This section describes the configuration parameters for to configure the limit the request at Undertow Queue for PCF Services such as SM, PDS, Binding, and Bulwark services.

Table 3-108 SM Service Helm Configurations

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
sm-service.undertow.queueRequestLimiter.enable	Enables Undertow Queue Request Limiter	Optional	false	CNC Policy, PCF, & PCRF	Added in 24.2.1
sm-service.undertow.queueRequestLimiter.discardPriority	The maximum acceptable request size by the Undertow queue.	Optional	0	CNC Policy, PCF, & PCRF	Added in 24.2.1
sm-service.undertow.queueRequestLimiter.maxAcceptRequestCount	The discard priority of the request.	Optional	5000	CNC Policy, PCF, & PCRF	Added in 24.2.1

Sample Helm Configuration for SM Service:

```
sm-service:
  serverHttpEnableBlockingReadTimeout: true
  undertow:
    queueRequestLimiter:
      enable: true
      discardPriority: 0
      maxAcceptRequestCount: 5000
```

Table 3-109 PolicyDS Service Helm Configurations

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
policyds.undertow.queueRequestLimiter.enable	Enables Undertow Queue Request Limiter	Optional	false	CNC Policy, PCF, & PCRF	Added in 24.2.1
policyds.undertow.queueRequestLimiter.discardPriority	The maximum acceptable request size by the Undertow queue.	Optional	0	CNC Policy, PCF, & PCRF	Added in 24.2.1

Table 3-109 (Cont.) PolicyDS Service Helm Configurations

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
policyds.undertow.queueRequestLimiter.maxAcceptRequestCount	The discard priority of the request.	Optional	5000	CNC Policy, PCF, & PCRF	Added in 24.2.1

Sample Helm Configuration for PolicyDS Service:

```
policyds:
  serverHttpEnableBlockingReadTimeout: true
  undertow:
    queueRequestLimiter:
      enable: true
      discardPriority: 0
      maxAcceptRequestCount: 5000
```

Table 3-110 Binding Service Helm Configurations

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
binding.undertow.queueRequestLimiter.enable	Enables Undertow Queue Request Limiter	Optional	false	CNC Policy, PCF, & PCRF	Added in 24.2.1
binding.undertow.queueRequestLimiter.discardPriority	The maximum acceptable request size by the Undertow queue.	Optional	0	CNC Policy, PCF, & PCRF	Added in 24.2.1
binding.undertow.queueRequestLimiter.maxAcceptRequestCount	The discard priority of the request.	Optional	5000	CNC Policy, PCF, & PCRF	Added in 24.2.1

Sample Helm Configuration for Binding Service:

```
binding:
  serverHttpEnableBlockingReadTimeout: true
  undertow:
    queueRequestLimiter:
      enable: true
      discardPriority: 0
      maxAcceptRequestCount: 5000
```

Table 3-111 Bulwark Service Helm Configurations

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release
bulwark.undertow.queueRequestLimiter.enable	Enables Undertow Queue Request Limiter	Optional	false	CNC Policy, PCF, & PCRF	Added in 24.2.1
bulwark.undertow.queueRequestLimiter.discardPriority	The maximum acceptable request size by the Undertow queue.	Optional	0	CNC Policy, PCF, & PCRF	Added in 24.2.1
bulwark.undertow.queueRequestLimiter.maxAcceptRequestCount	The discard priority of the request.	Optional	5000	CNC Policy, PCF, & PCRF	Added in 24.2.1

Sample Helm Configuration for Bulwark Service:

```
bulwark:
  serverHttpEnableBlockingReadTimeout: true
  undertow:
    queueRequestLimiter:
      enable: true
      discardPriority: 0
      maxAcceptRequestCount: 5000
```

3.46 Configuring Kafka for NF message feed

This section describes the parameters that are required to configure Kafka for NF message feed.

Table 3-112 Parameters for Message Feed Configuration for Kafka

Parameter	Description	Mandatory/Optional Parameter	Default Value
global.nfType	Identifies the type of producer NF.	Optional	PCF
global.nfInstanceId	Identifies the producer NF instance.	Optional	6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c
global.nfFqdn	Identifies the producer NF fqdn.	Optional	PCF-d5g.oracle.com

4
Enabling LoadBalancer with MetalLB

Oracle Communications Cloud Native Environment (OCCNE) have MetalLB installed, and free external IPs are already configured under MetalLB. This section is applicable only for CNC Policy and cnPCRF.

Perform the following steps to enable LoadBalancer to specific services.

> **Note**
>
> MetalLB configuration is supported only from OCCNE 1.4.

> **Note**
>
> In the CNC Policy and cnPCRF namespaces, only diam-gateway service and cm service with GUI page requires loadbalancer setting with accessible external IP.

4.1 Updating diam-gateway Service

To update diam-gateway service:

1. Login to Kubernetes cluster master node using ssh command.
2. Run the following command to edit svc yaml file for diam-gateway:

   ```
   kubectl edit svc diam-gateway-service -n PCRF_NAME_SPACE
   ```

Table 4-1 Variables

Variable Name	Description
diam-gateway-service	The name of diam-gateway service in setup.
PCRF_NAME_SPACE	The --namespace value used in helm install command.

Following is an sample content that displays in diam-gateway edit window.

```
1 # Please edit the object below. Lines beginning with a '#' will be
ignored,
 2 # and an empty file will abort the edit. If an error occurs while
saving this file will be
 3 # reopened with the relevant failures.
 4 #
 5 apiVersion: v1
 6 kind: Service
 7 metadata:
 8   creationTimestamp: 2019-06-02T13:06:11Z
 9   labels:
```

```
10     category: common
11     io.kompose.service: <PCRF_NAME>-pcrf-diam-gateway-service
12   name: <PCRF_NAME>-pcrf-diam-gateway-service
13   namespace: <PCRF_NAME_SPACE>
14   resourceVersion: "21624671"
15   selfLink: /api/v1/namespaces/<PCRF_NAME_SPACE>/services/<PCRF_NAME>-
pcrf-diam-gateway-service
16   uid: 31a4b13f-8537-11e9-81c8-0010e08b3a8e
17 spec:
18   clusterIP: 10.20.37.37
19   externalTrafficPolicy: Cluster
20   ports:
21   - name: diameter
22     nodePort: 32592
23     port: 3868
24     protocol: TCP
25     targetPort: 3868
26   - name: http
27     nodePort: 31301
28     port: 8080
29     protocol: TCP
30     targetPort: 8080
31   selector:
32     io.kompose.service: <PCRF_NAME>-pcrf-diam-gateway-service
33   sessionAffinity: None
34   type: NodePort
35 status:
36   loadBalancer: {}
```

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

   `annotations:`

   `metallb.universe.tf/address-pool: `*`ADDRESS_POOL_NAME`

   > **Note**
   >
   > - As per user MetalLB setting, you should select an appropriate pool name to replace the variable, *ADDRESS_POOL_NAME*
   > - *annotation:* line must be kept vertical align with line 16, while following line, *metallb.universe.tf/address-pool: ADDRESS_POOL_NAME* must be kept vertical align with line 10. If vertical align restriction failed to follow this rule, the svc yaml file update may fail.

4. Quit vim editor and save changes.

Updating cm-service

Follow the same process to update svc yaml for *PCRF_NAME* -pcrf-cm-service.

4.2 Updating Ingress Gateway Service

To update Ingress gateway service:

1. Login to Kubernetes cluster master node using ssh command.

2. Run the following command to edit svc yaml file for ingress gateway:

   `kubectl edit svc ingress-gateway-service -n PCF_NAME_SPACE`

   Table 4-2 Variables

Variable Name	Description
ingress-gateway-service	The name of ingress-gateway service in setup.
PCF_NAME_SPACE	The --namespace value used in Helm install command.

3. The following MetalLB configuaration lines need to be added in the *annotation:* of *metadata*:

   `annotations:`

   `metallb.universe.tf/address-pool: ADDRESS_POOL_NAME`

   `metallb.universe.tf/allow-shared-ip: sharedip`

   > **Note**
   >
   > As per user MetalLB setting, you should select an appropriate pool name to replace the variable, *ADDRESS_POOL_NAME*

   > **Note**
   >
   > The parameter `global.metalLbIpAllocationEnabled` must be enabled for metallb IP allocation.

4. Quit vim editor and save changes.

5
Upgrading Policy

This chapter provides information about upgrading Oracle Communications Cloud Native Core, Converged Policy (Policy) deployment to the latest release. It is recommended to perform Policy upgrade in a specific order. For more information about the upgrade order, see *Oracle Communications Cloud Native Core, Solution Upgrade Guide*.

> **Note**
>
> - In a georedundant deployment, perform the steps explained in this section on all the georedundant sites.
> - For Policy georedundant deployments, all the georedundant sites are expected to be upgraded to the common version before any individual site of GR deployment is planned for additional upgrade.
>
> For example, In a three-site Policy deployment, all the 3 sites are at the same release version as N. During site upgrade, site 1 and site 2 are upgraded to N+1 version, and site 3 is not upgraded yet. At this state, before upgrading site 3 to N+1/N+2, upgrading site 1 or site 2 from N+1/N+2 version to higher version is not supported as site3 in the georedundant environment is still not upgraded to N+1/N+2.
>
> For more information about the cnDBTier georedundant deployments, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
>
> For more information about the CNC Console georedundant deployments, see *Oracle Communications Cloud Native Core, CNC Console Installation, Upgrade, and Fault Recovery Guide*.

5.1 Supported Upgrade Paths

The following table lists the supported upgrade paths for Policy:

Table 5-1 Supported Upgrade Paths

Source Release	Target Release
24.2.x	24.3.0
24.1.x	24.3.0

> **Note**
>
> Policy must be upgraded before upgrading cnDBTier.

5.2 Upgrade Strategy

Policy supports in-service upgrade. The supported upgrade strategy is `RollingUpdate`. The rolling update strategy is a gradual process that allows you to update your Kubernetes system with only a minor effect on performance and no downtime. The advantage of the rolling update strategy is that the update is applied Pod-by-Pod so the greater system can remain active.

> **Note**
>
> It is recommended to perform in-service upgrade during maintenance window where the recommended traffic rate is 25% of the configured traffic or below. We also expect the traffic failure to stay below 5% during the upgrade and fully recover post upgrade.

The following engineering configuration parameters are used to define upgrade strategy:

- `upgradeStrategy` parameter indicates the update strategy used in Policy.
- `maxUnavailable` parameter determines the maximum number of pods that will be unavailable during upgrade.
  For more information on `maxUnavailable` for each microservices

  refer PodDisruptionBudget Configuration section.

> **Note**
>
> When Policy is deployed with OCCM, follow the specific upgrade sequence as mentioned in the *Oracle Communications, Cloud Native Core Solution Upgrade Guide*.

> **Note**
>
> During an in-service Helm upgrade transient errors may occur which are typically resolved by the Network Element's retry mechanism. It is done either by using a different available pod on the same site or by retrying at another site.
>
> It is recommended is to execute in-service Helm upgrade during maintenance window or low traffic period to minimize any service impact.

5.3 Preupgrade Tasks

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

1. Keep current custom_values.yaml file as backup.
2. Update the new custom_values.yaml file for target Policy release. For details on customizing this file, see Customizing Policy.
3. While upgrading Policy from a base version where database slicing was introduced to SM service (Policy 22.4.0), Usage Monitoring service (Policy 23.4.0) or PCRF Core (Policy 24.2.0) or at a version above from where database slicing was introduced, manually create

the sliced table for these services if table slicing was not previously enabled for that service and if table slicing is enabled during upgrade.

For example, if the base version is Policy 23.3.0 and Policy is installed with no slicing enabled for any of the services. To enable slicing for Usage Monitoring service and upgrade to any latest version (say 24.1.0), manually create sliced tables of UmContext database.

Name of the tables must be in the format: <tableName>_1, <tableName>_2...<tableName>_n. Number of sliced tables to be created must be slicing count -1.

For example, if umContextTableSlicingCount is 8 then following sliced tables need to be manually created before upgrade - UmContext_1, UmContext_2, UmContext_3 ... UmContext_7

4. In order to enable database slicing after an upgrade from one version to the same version, since the database hooks are not executed, manually create the database slices on database directly.

   a. Run the following command to create the database slices:

```
CREATE TABLE `gxsession_<slice_number>` (
  `id` varchar(255) NOT NULL,
  `value` varchar(20000) DEFAULT NULL,
  `nai` varchar(255) DEFAULT NULL,
  `ipv4` varchar(20) DEFAULT NULL,
  `ipv6` varchar(50) DEFAULT NULL,
  `e164` varchar(20) DEFAULT NULL,
  `imsi` varchar(20) DEFAULT NULL,
  `imei` varchar(20) DEFAULT NULL,
  `ipd` varchar(255) DEFAULT NULL,
  `updated_timestamp` bigint unsigned DEFAULT '0',
  `lastaccesstime` datetime DEFAULT CURRENT_TIMESTAMP,
  `siteid` varchar(128) DEFAULT NULL,
  `compression_scheme` tinyint unsigned DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_gxsession_ipv4` (`ipv4`),
  KEY `idx_gxsession_e164` (`e164`),
  KEY `idx_gxsession_nai` (`nai`),
  KEY `idx_gxsession_ipv6` (`ipv6`),
  KEY `idx_gxsession_imsi` (`imsi`),
  KEY `idx_gxsession_imei` (`imei`),
  KEY `idx_gxsession_ipd` (`ipd`),
  KEY `idx_audit_datetime` (`lastaccesstime`)
) ENGINE=ndbcluster DEFAULT CHARSET=latin1
COMMENT='NDB_TABLE=NOLOGGING=1'
```

   Here `<slice_number>` refers to the number of slices to be created.

   For example, if the number of slices to be created is 3, run the above command two times, first by replacing `gxsession_<slice_number>` with `gxsession_1`, and then with `gxsession_2`.

   b. Configure the database slicing feature with the advanced setting (DISTRIBUTE_GX_TRAFFIC_USING_TABLE_SLICING) and the deployment variable for GX_SESSION_TABLE_SLICING_COUNT according to the number of slices that you manually created. For more information see *Configurable Parameters for Database Slicing* table in Database Load Balancing Configuration.

> **Note**
>
> In case of rollback to a previous version of Policy software, all the sessions that are saved in the slices will remain in those tables and will not be moved to the main table.
>
> For the upgrade process, it is recommended to have the new policy installation in the cluster.

5. Before starting the upgrade, take a manual backup of Policy REST based configuration. This helps if preupgrade data has to be restored.

> **Note**
>
> For Rest API configuration details, see *Oracle Communications Cloud Native Core, Converged Policy REST Specification Guide*.

6. Before upgrading, perform sanity check using Helm test. See the Performing Helm Test section for the Helm test procedure.
7. Before Upgrading to release 24.2.0 from any previous release, make sure there are no entries for UDR Connector and CHF Connector in `ReleaseConfig` Table of `<occnp_release> database (“<occnp_release>” is the database name)`. Following are the commands for deleting the entries:

```
DELETE FROM `<occnp_release>`.`ReleaseConfig` WHERE  CfgKey =
‘public.hook.chf-connector’;
DELETE FROM `<occnp_release>`.`ReleaseConfig` WHERE  CfgKey =
‘public.hook.udr-connector’;
```

5.4 Upgrade Tasks

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

Helm Upgrade

Upgrading an existing deployment replaces the running containers and pods with new containers and pods. If there is no change in the pod configuration, it is not replaced. Unless there is a change in the service configuration of a microservice, the service endpoints remain unchanged.

Upgrade Procedure

> **⚠ Caution**
>
> - Stop the provisioning traffic before you start the upgrade procedure.
> - Do not perform any configuration changes during the upgrade.
> - Do not exit from `helm upgrade` command manually. After running the `helm upgrade` command, it takes some time (depending upon the number of pods to upgrade) to upgrade all the services. In the meantime, you must not press "`ctrl+c`" to come out from `helm upgrade` command. It may lead to anomalous behavior.

1. Untar the latest Policy package and if required, re-tag and push the images to registry. For more information, see Downloading Policy package and Pushing the Images to Customer Docker Registry.
2. Modify the `occnp_custom_values_24.3.0.yaml` file parameters as per site requirement.
3. Do not change the `nfInstanceId` configuration for the site. In case of multisite deployments, configure `nfInstanceId` uniquely for each site.
4. Assign appropriate values to `core_services` in the `appInfo` configuration based on `policy` Mode.
5. Run the following command to upgrade an existing Policy deployment:

   > **ⓘ Note**
   >
   > If you are upgrading an existing Policy deployment with georedundancy feature enabled, ensure that you configure `dbMonitorSvcHost` and `dbMonitorSvcPort` parameters before running `helm upgrade`. For more information on the parameters, see

   - Using local Helm chart:

     ```
     helm upgrade <release_name> <helm_chart> -f
     <occnp_customized_values.yaml> --namespace <namespace>
     ```

     Where,

     `<release_name>` is the Policy release name.

     `<helm_chart>` is the Helm chart.

     `<policy_customized_values.yaml>` is the latest custom-values.yaml file. For example, `occnp_custom_values_24.3.0.yaml`

     `<namespace>` is namespace of Policy deployment.

     For example:

     ```
     helm upgrade occnp occnp-pkg-24.3.0.0.0.tgz -f
     occnp_custom_values_24.3.0.yaml --namespace occnp
     ```

- Using chart from Helm repo:

  ```
  helm upgrade <release_name> <helm_repo/helm_chart> --version
  <chart_version> -f <policy_customized_values.yaml> --namespace
  <namespace>
  ```

  Where,

  `<release_name>` is the Policy release name.

  `<helm_repo/helm_chart>` is the Helm repository for Policy.

  `<policy_customized_values.yaml>` is the latest custom-values.yaml file. For example, `occnp-24.3.0-custom-values-occnp.yaml`

  `<namespace>` is namespace of Policy deployment.

  For example:

  ```
  helm upgrade occnp occnp-helm-repo/occnp --version 24.3.0 -f
  occnp_custom_values_24.3.0.yaml --namespace occnp
  ```

  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.
  - **timout duration**: If not specified, default value will be 300 (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.

> **Note**
>
> It is recommended not to use `--wait` and `--atomic` parameters along with helm upgrade as this might result in upgrade failure.

Note

The following warnings must be ignored for policy upgrade on 24.1.0, 24.2.0, and 24.3.0 CNE:

```
helm upgrade <release-name> -f <custom.yaml> <tgz-file> -n
<namespace>
W0301 15:46:11.144230 2082757 warnings.go:70]
spec.template.spec.containers[0].env[21]: hides previous definition
of "PRRO_JDBC_SERVERS"
W0301 15:46:48.202424 2082757 warnings.go:70]
spec.template.spec.containers[0].ports[3]: duplicate port
definition with spec.template.spec.containers[0].ports[1]
W0301 15:47:25.069699 2082757 warnings.go:70]
spec.template.spec.containers[0].ports[3]: duplicate port
definition with spec.template.spec.containers[0].ports[2]
W0301 15:47:43.260912 2082757 warnings.go:70]
spec.template.spec.containers[0].ports[4]: duplicate port
definition with spec.template.spec.containers[0].ports[2]
W0301 15:47:51.457088 2082757 warnings.go:70]
spec.template.spec.containers[0].ports[4]: duplicate port
definition with spec.template.spec.containers[0].ports[2]
Release "<release-name>" has been upgraded. Happy Helming!
NAME: <release-name>
LAST DEPLOYED: <Date-Time>
NAMESPACE: <namespace>
STATUS: deployed
REVISION: <N>
```

6. Run the following command to check the status of the upgrade:

   ```
   helm status <release_name> --namespace <namespace>
   ```

   Where,

   `<release_name>` is the Policy release name.

   `<namespace>` is namespace of Policy deployment.

   For example:

   ```
   helm status occnp --namespace occnp
   ```

7. Perform sanity check using Helm test. See the Performing Helm Test section for the Helm test procedure.
8. If the upgrade fails, see "*Upgrade or Rollback Failure*" in *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide*.

Note

If Usage Monitoring Service is enabled during upgrade to 24.2.1, then the log level must be set to WARN in the CNC Console for the Usage Management Service.

Note

To automate the lifecycle management of the certificates through OCCM, you can migrate certificates and keys from Policy to OCCM. For more information, see "Introducing OCCM in an Existing NF Deployment" in Oracle Communications Cloud Native Core, Certificate Management User Guide.

You can remove Kubernetes secrets if the current version of Policy does not use that secret by checking the `occnp_custom_values.yaml` file. Before deleting, please make sure that there is no plan to rollback to the Policy version which uses these secrets. Otherwise Rollback will fail.

After the upgrade http_server_requests_seconds metric with dimension {pod=~".*ueservice.*}" for UE service is replaced with occnp_ueservice_overall_processing_time_seconds and http_server_requests_seconds metric with dimension {pod=~".*amservice.*}" for AM service is replaced with occnp_amservice_overall_processing_time_seconds. Make sure to use the new metrics:

- For UE service:
  - occnp_ueservice_overall_processing_time_seconds_max instead of http_server_requests_seconds_max
  - occnp_ueservice_overall_processing_time_seconds_sum instead of http_server_requests_seconds_sum
  - occnp_ueservice_overall_processing_time_seconds_count instead of http_server_requests_seconds_count
- For AM service:
  - occnp_amservice_overall_processing_time_seconds_max instead of http_server_requests_seconds_max
  - occnp_amservice_overall_processing_time_seconds_sum instead of http_server_requests_seconds_sum
  - occnp_amservice_overall_processing_time_seconds_count instead of http_server_requests_seconds_count

For more details, see *UE Service Metrics* and *AM Service Metrics* sections in *Oracle Communications Cloud Native Core, Converged Policy User Guide*.

5.5 MIB Management

toplevel.mib and *POLICY-ALARM-MIB.mib* are two MIB files which are used to generate the traps. You must update these files along with the Alert file in order to fetch the traps in their environment. The MIB files are managed by SNMP manager.

Note

policy-alarm-mib.mib file has been replaced by *POLICY-ALARM-MIB.mib* file.

6
Rolling Back Policy

This chapter provides information about rolling back Oracle Communications Cloud Native Core, Converged Policy (Policy) deployment to the previous release. It is recommended to perform Policy rollback in a specific order. For more information about the rollback order, see *Oracle Communications Cloud Native Core Solution Upgrade Guide*.

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

6.1 Supported Rollback Paths

The following table lists the supported rollback paths for Policy:

Table 6-1 Supported Rollback Paths

Source Release	Target Release
24.3.0	24.2.x
24.3.0	24.1.x

> **Note**
>
> If georedundancy feature was disabled before upgrading to 24.3.0, 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.

6.2 Rollback Tasks

To roll back from Policy 24.3.0 to a previous version:

Note

- No configuration should be performed during rollback.
- **Caution**

  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 Policy pod is in the failed state.

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

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

   Where,

   - `<release_name>` is the release name used by the Helm command.
   - `<release_namespace>` is the namespace where Policy is deployed.

   For example:

   ```
   helm history occnp --namespace occnp
   ```

2. If you are rolling back from Policy 23.2.4, perform the following tasks for audit registration to be successful and audit cycle to continue as expected:

   a. Run the following command to find if JSON element `"handleNullAsStale":false` exists in `audit_req_data` column for a service.

      ```
      select * from AuditRegistrations\G
      ```

   b. If the JSON element exits, copy the JSON object string from column `audit_reg_data` and update it with the JSON element `"handleNullAsStale":false` or `"handleNullAsStale":true` element being removed and then use the updated JSON string for the column `audit_reg_data` in the below UPDATE command.

      ```
      UPDATE occnp_audit_service.auditregistrations SET
      audit_req_data='{audit_req_data}' where service_name='{servicename}';
      ```

      `{audit_req_data}` should be the updated column data and `{servicename}` should be the value of the column `service_name`.

   c. Run the following command to verify that the element is no longer in the JSON value.

      ```
      select * from AuditRegistrations\G
      ```

   d. Verify that the audit process is functioning as expected.

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

   ```
   helm rollback <release_name> <revision_number> --namespace
   <release_namespace>
   ```

   Where, `<revision_number>` is the release number to which Policy needs to be rolled back.

   For example:

   ```
   helm rollback occnp 1 --namespace occnp
   ```

   Note

   The following warnings must be ignored for policy rollback on 24.2.0 and 24.3.0 CNE:

   ```
   helm rollback <release-name> <revision_number> --namespace
   <release_namespace>
   W0801 11:51:20.139886 3453570 warnings.go:70]
   spec.template.spec.containers[0].env[21]: hides previous definition
   of "PRRO_JDBC_SERVERS"
   W0801 11:51:33.330235 3453570 warnings.go:70]
   spec.template.spec.containers[0].ports[3]: duplicate port
   definition with spec.template.spec.containers[0].ports[1]
   W0801 11:51:54.739481 3453570 warnings.go:70]
   spec.template.spec.containers[0].ports[3]: duplicate port
   definition with spec.template.spec.containers[0].ports[2]
   W0801 11:52:01.125243 3453570 warnings.go:70]
   spec.template.spec.containers[0].ports[4]: duplicate port
   definition with spec.template.spec.containers[0].ports[2]
   W0801 11:52:05.530639 3453570 warnings.go:70]
   spec.template.spec.containers[0].ports[4]: duplicate port
   definition with spec.template.spec.containers[0].ports[2]
   Rollback was a success! Happy Helming!
   ```

4. If the rollback fails, see **Upgrade or Rollback Failure** in *Oracle Communications Cloud Native Core, Converged Policy Troubleshooting Guide.*

7 Uninstalling Policy

This chapter provides information about uninstalling Oracle Communications Cloud Native Core, Converged Policy (Policy).

7.1 Uninstalling Policy using Helm

This chapter describes how to uninstall Policy using Helm.

To uninstall Policy, run the following command:

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

Where,

`<helm-release>` is a name provided by the user to identify the helm deployment.

`<release-namespace>` is a name provided by the user to identify the namespace of Policy deployment.

For example:

```
helm uninstall occnp --namespace occnp
```

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 occnp
```

> **Note**
>
> The following warnings must be ignored for policy uninstallation on 24.1.0, 24.2.0, and 24.3.0 CNE:
>
> ```
> helm uninstall <releasename> -n <namespace>
> W1025 02:22:44.254488 2521118 warnings.go:70] unknown field
> "spec.template.spec.resources"
> release "<Releasename>" uninstalled
> ```

> **Note**
>
> - When you uninstall both Policy and cnDBtier network functions, it is recommended to delete PVC (PersistentVolumeClaims) volumes of cnDBTier that were created at the time of installing cnDBtier. For more information on how to delete PVC volumes, see Deleting PVC Volumes.
> - If you are uninstalling only Policy network function, do not delete PVC volumes.

7.2 Deleting Kubernetes Namespace

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

To delete kubernetes namespace, run the following command:

```
kubectl delete namespace <release_namespace>
```

where, `<release_namespace>` is the deployment namespace used by the helm command.

For example:

```
kubectl delete namespace occnp
```

7.3 Removing Database Users

This section describes how to remove MySQL users.

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

Remove Privileged User:

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

For example:

```
DROP USER IF EXISTS ''occnpadminusr'@'%';
```

Remove Application User:

```
DROP USER IF EXISTS <Policy Application User Name>;
```

For example:

```
DROP USER IF EXISTS 'occnpusr'@'%';
```

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

7.4 Deleting PVC Volumes

This section describes how to delete a PVC volume.

Following is the procedure for to delete a PVC volume:

1. Get a list of the PVC volumes for the required cnDBTier namespace by running the following command:

   ```
   kubectl get pvc -n <namespace>
   ```

   where, `<namespace>` is the namespace of Policy deployment.

   For example:

   ```
   kubectl get pvc -n occnp
   ```

   Sample Output

   ```
   NAME                                STATUS
   VOLUME                                         CAPACITY   ACCESS MODES
   STORAGECLASS   AGE
   pvc-backup-ndbmtd-ndbmtd-0        Bound    pvc-56420da4-f70c-46fd-8f0a-
   c33ff50ddd98   3Gi        RWO            standard       26d
   pvc-backup-ndbmtd-ndbmtd-1        Bound    pvc-d7e3ef21-4161-40cc-
   abc9-756dfd3c0674   3Gi        RWO            standard       26d
   pvc-ndbappmysqld-ndbappmysqld-0   Bound    pvc-d76f0cbd-
   c0e5-48f1-9a83-1353dbee6c41   2Gi        RWO            standard       26d
   pvc-ndbappmysqld-ndbappmysqld-1   Bound    pvc-c31749df-ae98-48c1-
   a8a7-449535d0f60a   2Gi        RWO            standard       26d
   pvc-ndbmgmd-ndbmgmd-0             Bound    pvc-765a9a4b-726d-43fe-af91-
   a3a6c85ab321   1Gi        RWO            standard       26d
   pvc-ndbmgmd-ndbmgmd-1             Bound    pvc-71b5877e-a753-4fac-
   b995-5fdf2d465af8   1Gi        RWO            standard       26d
   pvc-ndbmtd-ndbmtd-0               Bound    pvc-e0c5f263-d5d3-4f18-
   bb11-44ceaa6a2305   3Gi        RWO            standard       26d
   pvc-ndbmtd-ndbmtd-1               Bound    pvc-d03f799d-abe2-4b71-96c4-
   c98cacdbeaba   3Gi        RWO            standard       26d
   pvc-ndbmysqld-ndbmysqld-0         Bound    pvc-
   d80321a9-5239-4e34-9bcf-11053c9de5ef   2Gi        RWO
   standard       26d
   pvc-ndbmysqld-ndbmysqld-1         Bound    pvc-545e0f35-
   ad20-4c24-927a-6f1e49b06cc9   2Gi        RWO            standard       26d
   ```

2. Delete all the PVC volumes in the cnDBTier namespace by running the following command:

   ```
   kubectl -n <namespace> delete pvc <pvc_name>
   ```

   Example:

   ```
   kubectl -n p1 delete pvc pvc-backup-ndbmtd-ndbmtd-0
   kubectl -n p1 delete pvc pvc-backup-ndbmtd-ndbmtd-1
   ```

```
kubectl -n p1 delete pvc pvc-ndbappmysqld-ndbappmysqld-0
kubectl -n p1 delete pvc pvc-ndbappmysqld-ndbappmysqld-1
kubectl -n p1 delete pvc pvc-ndbmgmd-ndbmgmd-0
kubectl -n p1 delete pvc pvc-ndbmgmd-ndbmgmd-1
kubectl -n p1 delete pvc pvc-ndbmtd-ndbmtd-0
kubectl -n p1 delete pvc pvc-ndbmtd-ndbmtd-1
kubectl -n p1 delete pvc pvc-ndbmysqld-ndbmysqld-0
kubectl -n p1 delete pvc pvc-ndbmysqld-ndbmysqld-1
```

7.5 Uninstalling Site in Georedundant Deployment

This chapter describes how to uninstall a site (except the last site) when Policy is deployed in georedundant setup.

Deleting entries of unique databases
For georedundant deployment, run the following command to delete entries of unique databases from the `occnp_release` database:

```
delete from ReleaseConfig where SiteId='7c4f7f05-ffdd-408f-ba78-b2c4dc83b1fd'
AND CfgKey='public.hook.auditservice';
delete from ReleaseConfig where SiteId='7c4f7f05-ffdd-408f-ba78-b2c4dc83b1fd'
AND CfgKey='public.hook.cmservice';
delete from ReleaseConfig where SiteId='7c4f7f05-ffdd-408f-ba78-b2c4dc83b1fd'
AND CfgKey='public.hook.configserver';
```

Cleaning up NDB Replication Table
In addition, clean up the entries in "mysql.ndb_replication" table by running the following commands

1. Select cast(db as char), cast(table_name as char), cast(conflict_fn as char) from mysql.ndb_replication:

   ```
   mysql> select cast(db as char), cast(table_name as char), cast(conflict_fn
   as char) from mysql.ndb_replication;
   ```

   Sample Output

   ```
   +-----------------+-------------------------
   +---------------------------------------+
   | cast(db as char) | cast(table_name as char) | cast(conflict_fn as
   char)              |
   +-----------------+-------------------------
   +---------------------------------------+
   | occnp_binding    | dependentcontextbinding  |
   NDB$MAX_DELETE_WIN(lastModifiedTime)   |
   | occnp_pcrf_core  | rxsession                |
   NDB$MAX_DELETE_WIN(updated_timestamp)  |
   | occnp_pcf_am     | AmPolicyAssociation      |
   NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
   | occnp_pcf_sm     | AppSession               |
   NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
   | occnp_pcrf_core  | sessioncorrelationreg    |
   NDB$MAX_DELETE_WIN(updated_timestamp)  |
   | occnp_pcrf_core  | gxsession                |
   ```

```
NDB$MAX_DELETE_WIN(updated_timestamp)  |
| occnp_binding    | contextbinding           |
NDB$MAX_DELETE_WIN(lastModifiedTime)   |
| occnp_policyds   | pdsprofile               |
NDB$MAX_DELETE_WIN(last_modified_time) |
| occnp_pcf_ue     | UePolicyAssociation      |
NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
| occnp_policyds   | pdssubscriber            |
NDB$MAX_DELETE_WIN(last_modified_time) |
| occnp_pcf_sm     | SmPolicyAssociation      |
NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
+-----------------+-------------------------
+---------------------------------------+
11 rows in set (0.01 sec)
```

If the output for this command shows Policy databases as shown in the sample output, then perform Step 2.

2. Delete from mysql.ndb_replication:

```
delete from mysql.ndb_replication where cast(db as char)="<database_name>";
```

Example

```
delete from mysql.ndb_replication where cast(db as char)="occnp_pcf_am";
```

7.6 Uninstalling Last Site in Georedundant Deployment

This chapter describes how to uninstall the last site when Policy is deployed in georedundant setup. The same steps can be performed to uninstall Policy in standalone deployment.

7.6.1 Cleaning up NDB Replication Table

This chapter describes how to cleanup NDB replication table while uninstalling CNC Policy.

To clean up the entries in "mysql.ndb_replication" table, run the following commands:

1. Select cast(db as char), cast(table_name as char), cast(conflict_fn as char) from mysql.ndb_replication:

```
mysql> select cast(db as char), cast(table_name as char), cast(conflict_fn
as char) from mysql.ndb_replication;
```

Sample Output

```
+-----------------+-------------------------
+---------------------------------------+
| cast(db as char) | cast(table_name as char) | cast(conflict_fn as
char)              |
+-----------------+-------------------------
+---------------------------------------+
| occnp_binding    | dependentcontextbinding  |
NDB$MAX_DELETE_WIN(lastModifiedTime)   |
| occnp_pcrf_core  | rxsession                |
```

```
NDB$MAX_DELETE_WIN(updated_timestamp)  |
| occnp_pcf_am     | AmPolicyAssociation      |
NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
| occnp_pcf_sm     | AppSession               |
NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
| occnp_pcrf_core  | sessioncorrelationreg    |
NDB$MAX_DELETE_WIN(updated_timestamp)  |
| occnp_pcrf_core  | gxsession                |
NDB$MAX_DELETE_WIN(updated_timestamp)  |
| occnp_binding    | contextbinding           |
NDB$MAX_DELETE_WIN(lastModifiedTime)   |
| occnp_policyds   | pdsprofile               |
NDB$MAX_DELETE_WIN(last_modified_time) |
| occnp_pcf_ue     | UePolicyAssociation      |
NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
| occnp_policyds   | pdssubscriber            |
NDB$MAX_DELETE_WIN(last_modified_time) |
| occnp_pcf_sm     | SmPolicyAssociation      |
NDB$MAX_DELETE_WIN(UPDATED_TIMESTAMP)  |
+------------------+--------------------------
+----------------------------------------+
11 rows in set (0.01 sec)
```

If the output for this command shows CNC Policy databases as shown in the sample output, then perform Step 2.

2. Delete from mysql.ndb_replication:

```
delete from mysql.ndb_replication where cast(db as char)="<database_name>";
```

Example

```
delete from mysql.ndb_replication where cast(db as char)="occnp_pcf_am";
```

7.6.2 Cleaning up Databases

This chapter describes how to clean up databases when uninstalling CNC Policy.

> **Note**
>
> For georedundant deployment, run the commands provided in this section only if the site being uninstalled is the last site in the complete georedundant group.

To clean up database for the different microservices, run the following command:

```
DROP DATABASE IF EXISTS occnp_audit_service;
DROP DATABASE IF EXISTS occnp_config_server;
DROP DATABASE IF EXISTS occnp_pcf_am;
DROP DATABASE IF EXISTS occnp_pcf_sm;
DROP DATABASE IF EXISTS occnp_commonconfig;
DROP DATABASE IF EXISTS occnp_pcrf_core;
DROP DATABASE IF EXISTS occnp_release;
```

```
DROP DATABASE IF EXISTS occnp_binding;
DROP DATABASE IF EXISTS occnp_policyds;
DROP DATABASE IF EXISTS occnp_pcf_ue;
DROP DATABASE IF EXISTS occnp_cmservice;
DROP DATABASE IF EXISTS occnp_nrf_client;
DROP DATABASE IF EXISTS occnp_leaderPodDb;
DROP DATABASE IF EXISTS occnp_overload;
DROP DATABASE IF EXISTS occnp_pcf_nwdaf_agent;
```

8
Fault Recovery

This chapter provides information about fault recovery for Oracle Communications Cloud Native Core, Converged Policy (Policy) deployment.

8.1 Overview

You must take database backup and restore it either on the same or a different cluster. It uses the Policy database (MySQL NDB Cluster) to run any command or to follow any instructions.

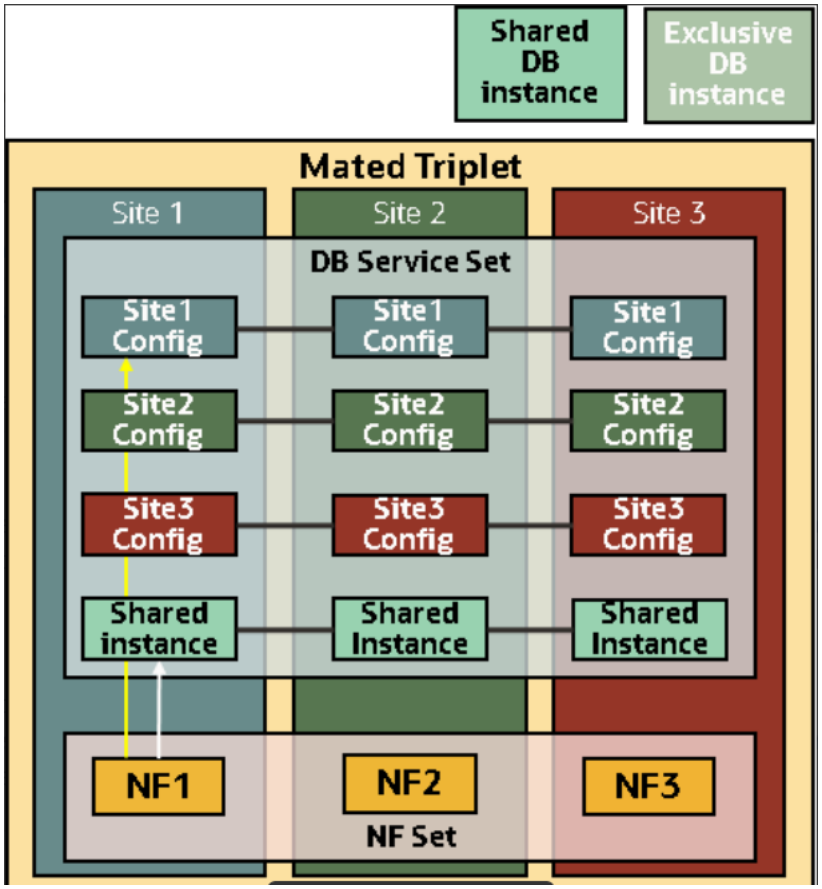
Database Model of CNC Policy

Policy database consists of the following two data types:

- **Configuration Data**: The configuration data is exclusive for a given site. Thus, an exclusive logical database is created and used by a site to store its configuration data. Using CNC Console and Configuration Management service, you can configure the data in the respective site only.
- **Session Data**: The session data is shared across sites. Thus, a common logical database is created and used by all sites. The data is replicated across sites to preserve and share session with mated sites. In case of cross sites messaging or a site failure, shared session data helps in continuity of service.

The following image shows the Policy database model in three different sites:

Figure 8-1 Database Model



8.2 Impacted Areas

The following table shares information about the impacted areas during Policy fault recovery:

Scenario	Requires Fault Recovery or re-install of CNE?	Requires Fault Recovery or re-install of cnDBTier?	Requires Fault Recovery or re-install of Policy?	Other
Scenario: Session Database Corruption	No	Yes Restoring cnDBTier from older backup is the only way to restore back to restore point.	No Only if cnDBTier credentials are changed.	All sites require Fault Recovery.
Scenario: Site Failure	Yes	Yes	Yes	NA

8.3 Prerequisites

Before performing any fault recovery procedure, ensure that the following prerequisites are met:

- cnDBTier must be in a healthy state and available on multiple sites along with Policy. To check the cnDBTier status, perform the following steps:
  1. Run the following command to ensure that all the nodes are connected:
     ```
     ndb_mgm> show
     ```
  2. Run the following command to check the pod status:
     ```
     kubectl get pods -n <namespace>
     ```
     If the pod status is `Running`, then the cnDBTier is in healthy state.
  3. Run the following command to check if the replication is up:
     ```
     mysql> show slave status\G
     ```
     In case there is any error, see *Fault Recovery* chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation and Upgrade Guide*.
  4. Run the following command to check which cnDBTier is having ACTIVE replication to take backup:
     ```
     select * from replication_info.DBTIER_REPLICATION_CHANNEL_INFO;
     ```
- Automatic backup must be enabled on cnDBTier. Enabling automatic backup helps in achieving the following:
  - Restore stable version of the network function database
  - Minimize significant loss of data due to upgrade 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 database information for a network function from one site to another
- The following files must be available for fault recovery:
  - Custom values file (`occnp-custom-values-<release_number>`)
  - Helm charts (`occnp-<release_number>.tgz`)
  - Secrets and Certificates
  - RBAC resources

> **Note**
>
> For details on enabling automatic backup, see *Fault Recovery* section in *Oracle Communications Cloud Native Core, cnDBTier (cnDBTier) Installation Guide*.

8.4 Fault Recovery Scenarios

This section describes the fault recovery procedures for various scenarios.

> **Note**
>
> This chapter describes scenario based procedures to restore Policy databases only. To restore all the databases that are part of cnDBTier, see *Fault Recovery* chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation and Upgrade Guide* available on My Oracle Support (MOS).

8.4.1 Scenario: Session Database Corruption

This section describes how to recover Policy when its session database corrupts.

When the session database corrupts, the database on all other sites can also corrupt due to data replication. It depends on the replication status after the corruption has occurred. If the data replication breaks due to database corruption, cnDBTier fails in either single or multiple sites (not all sites). And if the data replication is successful, database corruption replicates to all the cnDBTier sites and cnDBTier fails in all sites.

The fault recovery procedure covers following sub-scenarios:

- When DBTier Failed in All Sites

8.4.1.1 When DBTier Failed in All Sites

This section describes how to recover session database when successful data replication corrupts all the cnDBTier sites.

To recover session database, perform the following steps:

1. Uninstall Policy Helm charts on all sites. For more information about uninstalling Helm charts, see *Oracle Communication Cloud Native Core, Converged Policy Installation and Upgrade Guide* available on MOS.
2. Perform cnDBTier fault recovery procedure:
   a. Use auto-data backup file for restore procedure. For more information about DBTier restore, see *Fault Recovery* chapter in *Oracle Communications Cloud Native Core, cnDBTier Installation and Upgrade Guide* available on MOS.
3. Install Policy Helm charts. For more information about installing Helm charts, see *Oracle Communication Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide* available on MOS.

> **Note**
>
> You can also refer to the `custom-values.yaml` file used at the time of Policy installation for Helm charts installation.

8.4.2 Scenario: Site Failure

This section describes how to perform fault recovery when either one or many of your sites have software failure.

This section consists of the following:

- Single or Multiple Site Failure

8.4.2.1 Single or Multiple Site Failure

This scenario applies when one or more sites, and not all sites, have failed and there is a requirement to perform fault recovery. It is assumed that you have cnDBTier and Policy installed on multiple sites with automatic data replication and backup enabled.

> **Note**
>
> It is assumed that one of the cnDBTier is in healthy state.

To recover the failed sites, perform the following steps:

> **Note**
>
> Ensure that all the prerequisites mentioned are met.

1. Uninstall Policy. For more information, see the *Uninstalling CNC Policy* section in *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.
2. Install a new cluster by performing the Cloud Native Environment (CNE) installation procedure. For more information, see *Oracle Communications Cloud Native Core, Cloud Native Environment (CNE) Installation and Upgrade Guide* available on My Oracle Support.
3. 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 and Upgrade Guide*.
4. Perform DBTier fault recovery procedure:
   a. Perform DBTier 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 and Upgrade Guide*.
   b. Restore the database to new site by following the *Restore Database with Backup* procedure in *Oracle Communications Cloud Native Core, cnDBTier Installation and Upgrade Guide*.
5. Install Policy Helm charts. For more information on installing Helm charts, see *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade and Fault Recovery Guide*.

A

Resilient Policy Microservices with Kubernetes

Policy microservices pods run on Kubernetes cluster. Occasionally, Kubernetes pod disruptions may occur within the cluster, either from voluntary or involuntary causes. This results in pod/service failing, and the cluster experiences some outage resulting in discontinuity of the service. In order to mitigate these disruptions, the policy Kubernetes cluster and the services running on it are designed to be resilient by adapting the recommendations and strategies from Kubernetes framework. Thus, improving the availability, prevent/minimize the downtime and outages before they occur.

Described below are the various failure points that might occur in the policy cluster, and the resilience model that is adopted to handle it.

Table A-1 Failure point, Worker Node Failure

Failure Point: Worker Node failure
Recovery - Multiple pod By having multiple nodes in a cluster, it provides high availability by scheduling pods in different nodes, removing the single point of failure. By running multiple copies of policy services/pods reduces the chances of outages and service degradation. For more information about this functionality, see section Policy Services.
Recovery - Anti-Affinity rules The placement of the pod and any of its replicas can be controlled using Kubernetes pod affinity and anti-affinity rules. Pod anti-affinity rule is used to instruct Kubernetes not to co-locate pods on the same node. This avoids an outage due to the loss of a node. For more information about this functionality, see section Anti-affinity Approach to Assign Pods to Nodes.

Table A-2 Failure Point: Physical Server (Hosting Worker Node/s) failure

Failure Point: Physical Server (Hosting Worker Node/s) failure
Recovery - Pod Topology Spread (PTS) Pod topology spread constraints tells the Kubernetes scheduler how to spread pods across nodes in a cluster. It can be across nodes, zones, regions, or other user-defined topology domains. They allow users to use labels to split nodes into groups. Then users can specify pods using a label selector and indicate to the scheduler how evenly or unevenly those pods can be distributed.

Table A-3 Failure Point: Cluster needs to be upgraded or needs to shut down

Failure Point: Cluster needs to be upgraded or needs to shut down
Recovery - PodDisruptionBudget (PDB) Setting PDB ensures that the cluster have a sufficient number of available replicas, to keep it functioning even during maintenance. Using the PDB, we define a number (or percentage) of pods that can be terminated. With PDB configured, Kubernetes drains a node following the configured disruption schedule. New pods is deployed on other available nodes. This approach ensures Kubernetes schedules workloads in an optimal way while controlling the disruption based on PDB configuration. For more information about this functionality, see section PodDisruptionBudget Configuration.

Table A-3 (Cont.) Failure Point: Cluster needs to be upgraded or needs to shut down

Failure Point: Cluster needs to be upgraded or needs to shut down
Recovery - Terminate gracefully When a pod is evicted, it is gracefully terminated honoring the termination `gracePeriod` setting in the custom yaml file.

Table A-4 Failure Point: Pod/Application failure

Failure Point: Pod/Application failure
Recovery - Kubernetes Probes Kubernetes provides probes i.e health checks to monitor and act on the state of pods (Containers) and to make sure only healthy pods serve traffic. With help of probes, we can control when a pod should be deemed started, ready for service, or live to serve traffic. Kubernetes gives three types of health checks probes: • Liveness probes let Kubernetes know whether the application is running or not. • Readiness probes let Kubernetes know when the application is ready to serve traffic. • Startup probes let Kubernetes know whether the application has properly started or not.

Table A-5 Failure Point: High traffic Rate

Failure Point: High traffic Rate
Recovery - Horizontal Pod Auto-Scaling (HPA) When there is an increase or drop in the traffic, Kubernetes can automatically increase or decrease the pod replicas that serve the traffic. Horizontal scaling means that the response to increased load is to deploy more pods. If the load decreases, and the number of pods is above the configured minimum, the HorizontalPodAutoscaler instructs the work load resource to scale back down.

Table A-6 Any(including intra/inter-NF communication Failures)

Any(including intra/inter-NF communication Failures)
Recovery - All policy service support metrics/logs to capture the behavior.

Policy Microservices Resilience details

The criticality of the service failures is indicated as HIGH, MEDIUM and LOW, and they mean:

- **HIGH**- Service failure impacts the traffic, and it can not be handled successfully.
- **MEDIUM**- Service failure impacts the traffic, and it cannot be handled successfully by the default processing model.
- **LOW**- Service failure does not impact the traffic directly.

> **Note**
>
> The performance and capacity of the Policy system may vary based on the Call model, Feature/Interface configuration, underlying CNE and hardware environment, including but not limited to the complexity of deployed policies, policy table size , object expression and custom json usage in policy design.

Table A-7 Policy Kubernetes cluster Resiliency Details

Service Name	Multi-pod	Affinity/ Anti-affinityRule	HPA	PDB	PTS	Node Selector	Serviceability status tracking[1]	Criticality	Impact of Service loss/ failure	Overload Control/ Protection	Dependent service tracking and reporting
Alternate Route Service	Y	Y	0%/ 80 %	1	N	Y	Y	**HIGH**	On DNS-SRV based alternate routing is enabled: • Handles subsequent messages on failure of initial producer. • Handles notifications on failure of consumer. • SRV based lookup for NRF and SCP.	N	N
AM Service	Y	Y	0%/ 30 %	20 %	N	Y	Y	**HIGH**	The loss of this service leads to • AM call failures for a site. • NF will be marked as de-registered at NRF.	N	N
App-info	Y	Y	3%/ 80 %	20 %	N	Y	Y	**HIGH**	This service tracks status of all services and cnDbTier. This is used by services like: • Nrf-client for NF registration: On appInfo pod failure, Nrfclient uses the last known state fetched from AppInfo. However, if NrfClient pod also restarts, cache data is lost, NF service will be suspended at NRF. • Diameter-gateway to track readiness status of cnDbtier: On appInfo pod failure, Diameter-gateway uses the last known state fetched from AppInfo. However, if diameter-gateway pod also restarts, then it will fail to detect DB availability and will not be able to accept signaling traffic.	N	N

Table A-7 (Cont.) Policy Kubernetes cluster Resiliency Details

Service Name	Multi-pod	Affinity/ Anti-affinityRule	HPA	PDB	PTS	Node Selector	Serviceability status tracking[1]	Criticality	Impact of Service loss/ failure	Overload Control/ Protection	Dependent service tracking and reporting
Audit Service	N	Y	1%/ 60 %	50 %	--	N	Y	**LOW**	This service handles stale session cleanup and retry binding create operation. Loss of this service leads to large number of stale records and failure of retry binding sessions.	N	N
Binding Service	Y	Y	0%/ 60 %	20 %	N	Y	Y	**HIGH**	This service is responsible for creating binding with BSF. Failure of this service means failure of N5/Rx flows.	N	N
Bulwark Service	Y	Y	0%/ 60 %	0	N	Y	Y	**MEDIUM**	This service provides concurrency across various interfaces in policy. Failure of this service means there can be concurrency issues when processing requests for same subscriber over same/multiple interfaces.	N	N
CHF-Connector	Y	Y	0%/ 50 %	20 %	N	--	Y	**MEDIUM**	Failure of this service means • Spending limit flow with CHF will be impacted, hence spending counter based policies can not be enforced. • However SM session is created or updated without spendinglimit data.	N	N
CM Service	Y	Y	-- (fixed set of replicas)	20 %	N	Y	Y	**HIGH**	This service provides user interface to make configuration changes for policy (including common service configuration e.g. Ingress gateway, Egress gateway, NrfClient etc). If this service fails, other services will not be able to fetch the configuration data. Common services pods can continue to run with existing configurations, but it will get impacted on pod restart cases.	N	N

Table A-7 (Cont.) Policy Kubernetes cluster Resiliency Details

Service Name	Multi-pod	Affinity/ Anti-affinity Rule	HPA	PDB	PTS	Node Selector	Serviceability status tracking[1]	Criticality	Impact of Service loss/ failure	Overload Control/ Protection	Dependent service tracking and reporting
Config Server	Y	Y	7%/80%	20%	N	Y	Y	**HIGH**	This service is responsible for providing any configuration change to other services. Other services continue to work with existing configuration data, but container restart or pod scaling will lead to readiness failure, as they can not accept traffic without new configuration.	N	N
Diameter Connector	Y	Y	<unknown>/40%	20%	N	--	Y	**HIGH**	--	N	N
Diameter Gateway	Y	Y	-- (fixed set of replicas)	20%	N	Y	Y	**HIGH**	The failure of this services impacts all diameter related traffic in a site. • PCF mode: BSF/AF can perform alternate routing due to connectivity failure. • PCRF mode: PCEF and CSCF can perform alternate routing to select alternate site. • Egress flows e.g. RAR over Gx/Rx will be impacted.	Y Enforce overload control for backend services.	DB status is tracked (through appInfo) using helm configuration, to determine the readiness status of gateway pod.
Egress Gateway	Y	Y	0%/80%	1	N	Y	Y	**HIGH**	The loss of this service means • All Egress gateway flows over HTTP (i.e. UDR, CHF, BSF, SMF notification, AMF notifications, Nrf management and discovery flows) is impacted. • NRF marks site as "SUSPENDED" due to loss of HB.	N	N

Table A-7 (Cont.) Policy Kubernetes cluster Resiliency Details

Service Name	Multi-pod	Affinity/ Anti-affinityRule	HPA	PDB	PTS	Node Selector	Serviceability status tracking[1]	Criticality	Impact of Service loss/ failure	Overload Control/ Protection	Dependent service tracking and reporting
Igress Gateway	Y	Y	0%/ 80 %	1	N	Y	Y	**HIGH**	The loss of this service impacts connectivity from SCP, NRF or other peers for ingress flow. Hence it will indicate site failure to peers at network/transport level. **Note**: There shall not be signaling failure, if consumers perform alternate routing to alternate site to achieve session continuity.	Y - Enforce overload control for backend services. - Backend as well as IGW rate limiting support.	N
LDAP Gateway	--	--	0%/ 60 %	20 %	--	Y	--	--	--	--	--
Notifier Service	Y	Y	0%/ 60 %	20 %	N	Y	Y	**LOW**	This service is responsible for custom notifications. Since PRE to notifier is fire-and-forget, thus loss for such notifications shall not cause any functionality loss. There is no impact of 3gpp signaling.	N	N
NRF Client NF Discovery	Y	Y	0%/ 80 %	25 %	N	Y	Y	**MEDIUM**	The loss of this service means • On-demand discovery procedures is directly impacted. • Signaling flows are impacted. Policy performs on-demand discovery of UDR, thus failure will lead to missing subscriber profile information. **Note**: Based on configuration, SM/AM/UE may accept service requests.	N	N

Table A-7 (Cont.) Policy Kubernetes cluster Resiliency Details

Service Name	Multi-pod	Affinity/ Anti-affinityRule	HPA	PDB	PTS	Node Selector	Serviceability status tracking[1]	Criticality	Impact of Service loss/ failure	Overload Control/ Protection	Dependent service tracking and reporting
NRF Client NF Management	Y(Policy need to enable multi-pod support - currently set to 1)	Y	0%/80%	NA	N	Y	Y	**HIGH**	This service is responsible for NF profile registration with NRF. It also performs NRF HB and NRF health check functionality. Loss of this service for HB Timer interval, means that NRF can mark a given PCF instance as SUSPENDED. As soon as Nrf-mgmt pod becomes available, it will automatically refresh Nf's profile at NRF and bring site back to REGISTERED state (if NRF state was suspended).	N	N
Nwdaf Agent	--	--	0%/60%	20%	--	--	--	--	--	--	--
PCRF-Core	Y	Y	0%/40%	20%	N	Y	Y	**HIGH**	This service is responsible for all 4G PCRF flows. The failure of this service impacts all PCRF flows. Diameter peers can detect error response from diameter-gateway and can retry those sessions at alternate site.	N	N

Table A-7 (Cont.) Policy Kubernetes cluster Resiliency Details

Service Name	Multi-pod	Affinity/ Anti-affinityRule	HPA	PDB	PTS	Node Selector	Serviceability status tracking[1]	Criticality	Impact of Service loss/ failure	Overload Control/ Protection	Dependent service tracking and reporting
Perf-Info	Y	Y	-- (Fixed set of replicas)	20%	N	Y	Y	**MEDIUM**	This service is responsible to calculate load and overload level. This is used by services like: • Nrf-client for load reporting - When perfInfo pod is down, Nrfclient currently use the last known load level fetched from PerfInfo. However, if NrfClient pod also restarts, then it will loose its cache information and hence will report load level as zero. • Ingress gateway/ Diameter-gateway to track overload status of back end services. - When perfInfo pod is down, these services currently use the last known state reported by perfInfo.	N	N
Policy Data Source (PDS)	Y	Y	0%/ 60%	20%	N	Y	Y	**MEDIUM**	This service is responsible for UDR, LDAP, SOAP etc communication and caching subscriber/context information. Failure for this service means fail to handle subscriber data, which is crucial for policy signaling. Each core services (e.g. SM/AM/UE) even with missing data, can handle the service requests gracefully.	N	N
PRE (Policy Run Time)	Y	Y	0%/ 80%	20%	N	Y	Y	**MEDIUM**	This service is responsible for policy evaluation. Without this, core service will run their default policy evaluation and hence operator defined policies will not be applied.	N	N
PRE-Test	N	Y	--	--	N	--	Y	**LOW**	Test service for test projects.	N	N

Table A-7 (Cont.) Policy Kubernetes cluster Resiliency Details

Service Name	Multi-pod	Affinity/ Anti-affinity Rule	HPA	PDB	PTS	Node Selector	Serviceability status tracking[1]	Criticality	Impact of Service loss/ failure	Overload Control/ Protection	Dependent service tracking and reporting
Query Service	Y	Y	0%/ 80 %	20 %	N	Y	Y	**LOW**	The loss of this service means, operator will not be able to perform query/session viewers functionality. HA to provide better serviceability.	N	N
SM Service	Y	Y	0%/ 50 %	20 %	N	Y	Y	**HIGH**	This service is responsible for handling N7 and N5 requests. On loss of this service, PCF will not be able to handle signaling traffic from SMF.	N	N
SOAP Connector	--	--	0%/ 60 %	20 %	--	Y	--	--	--	--	--
UDR Connector	Y	Y	0%/ 50 %	20 %	N	--	Y	**MEDIUM**	This is a critical service for signaling flow with UDR. On loss of this service, it will have impact on signaling traffic. But using the configurations in core services, sessions can be processed without subscriber profile data.	N	N
UE Service	Y	Y	0%/ 30 %	20 %	N	Y	Y	**HIGH**	This service is responsible for handling UE policy flow. On failure of this service, PCF will not be able to handle AMF flow for UE policies.	N	N
Usage Monitoring Service	Y	Y	0%/ 80 %	20 %	N	Y	Y	**MEDIUM**	This service is responsible for usage monitoring and grant related functions. Failure of this service means usage monitoring functionality will be impacted. However upon this service failure. SM/ PCRF sessions will continue to function.	N	N

1. Service status tracking model:
   - AppInfo monitors state of policy services and its publish is available through **appinfo_service_running** metric.
   - Alert "`POLICY_SERVICES_DOWN`" will be raised, if service is down (i.e. 0 running pods for this service).

B
PodDisruptionBudget Configuration

PodDisruptionBudget (PDB) is a Kubernetes resource that allows to achieve high availability of scalable application services when the cluster administrators perform voluntary disruptions to manage the cluster nodes. PDB restricts the number of pods that are down simultaneously from voluntary disruptions. Defining PDB is helpful to keep the services running undisrupted when a pod is deleted accidentally or deliberately. PDB can be defined for high available and scalable Policy services such as SM Service, AM Service, User service, app-info, perf-info, etc.

It allows safe eviction of pods when a Kubernetes node is drained to perform maintenance on the node. It uses the default value of the `maxUnavailable` parameter specified in the Helm chart to determine the maximum number of pods that are unavailable during a voluntary disruption. For example, if `maxUnavailable` is 50%, the evictions are allowed until not more than 50% of the desired replicas are unhealthy.

> **Note**
>
> The performance and capacity of the Policy system may vary based on the Call model, Feature/Interface configuration, underlying CNE and hardware environment, including but not limited to the complexity of deployed policies, policy table size , object expression and custom json usage in policy design.

The following Policy services support PDB:

Table B-1 Default PodDisruptionBudget for Policy Microservices

Microservice	PDB Default Value (maxUnavailable)
alternate_route	1
am-service	20%
app-info	20%
audit-service	50%
binding service	20%
bulwark	0
cm-service	20%
config-server	20%
diam-connector	20%
diam-gateway	20%
ingress_gateway	1
egress_gateway	1
ldap-gateway	20%
notifier service	20%
nrf-client-discovery	25%
nrf-client-management	NA
nwdaf-agent	20%

Table B-1 (Cont.) Default PodDisruptionBudget for Policy Microservices

Microservice	PDB Default Value (maxUnavailable)
pcrf-core	20%
perf-info	20%
policyds	20%
pre	20%
pre-test	--
queryservice	20%
sm-service	20%
soap connector	20%
udr connector	20%
ue-service	20%
user-service	20%
usage-mon	20%
chf-connector	20%

C

Deployment Service Type Selection

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