Oracle® Communications

Cloud Native Core Policy Installation Guide

Release 1.7.3
F34930-01
September 2020

Oracle Communications Cloud Native Core Policy Installation Guide, Release 1.7.3

F34930-01

Contents

My Oracle Support

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

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

- For Technical issues such as creating a new Service Request (SR), select **1**.
- For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.
- For Hardware, Networking and Solaris Operating System Support, select **3**.

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

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

What's New in This Guide

New and updated information in this guide in Release 1.7.3:

- Added the Integrating Aspen with CNC Policy appendix to describe the procedure for integrating Aspen mesh with CNC Policy.
- Added the Additional Configurable Parameters for Aspen mesh section to include new configurable parameters in Helm for Aspen mesh.
- Updated the Installation Procedure chapter to install CNC Policy 1.7.3.
- Added theMigrating Data from Release 1.6.x to Release 1.7.x section to migrate data from release 1.6.x to release 1.7.x.
- Added the Upgrading CNC Policy (1.7.x to 1.7.3) appendix to describe the procedure for upgrading CNC Policy from 1.7.x to 1.7.3.
- Added the Downgrading Cloud Native Core Policy appendix to describe the roll back procedure from CNC Policy 1.7.3 to previous version.

1 Introduction

Oracle Communications Cloud Native Core Policy (CNC Policy) solution provides a standard policy design experience that allows you to craft and deploy, from scratch, the policies in production in minutes. 5G brings the policy design experience to the next level by providing flexibility, extensibility, modularization and assurance to rapidly deploy new policies and enable use cases faster. In addition, the overlap in functionality between PCF and cnPCRF (for example, need for a policy engine, policy design, Rx, similarity between Sy and Nchf_SpendingLimitControl, etc.), enables us to build micro-services that can be used to provide cnPCRF and PCF functionality. So, Cloud Native Core Policy (CNC Policy) solution provides the functionalities of both PCF and cnPCRF. Even though it is a unified policy solution, you can still deploy the PCF and cnPCRF entirely independently. In this release, Single Release Bundle provides the following deployment models:

- Converged Deployment (CNC Policy)
- PCF Deployment
- cnPCRF Deployment

You can select the deployment model by selecting the different custom yaml file in release site, for example:

Released Custom yaml File	Purpose
occnp-1.7.3-custom-values-occnp.yaml	This is the custom yaml file for converged installation.
occnp-1.7.3-custom-values-pcf.yaml	This is the custom yaml file for PCF installation.
occnp-1.7.3-custom-values-pcrf.yaml	This is the custom yaml file for cnPCRF installation.

You can download the required custom yaml files from OHC. For detailed procedure, see Customizing Cloud Native Core Policy.

The Cloud Native Core Policy is a functional element for policy control decision and flows based charging control functionalities. The CNC Policy provides the following functions:

- Policy rules for application and service data flow detection, gating, QoS, and flow based charging
- Access and Mobility Management related policies to the Access and Mobility Management Function (AMF)
- Provide UE Route Selection Policies (URSP) rules to UE via AMF
- Accesses subscription information relevant for policy decisions in a Unified Data Repository (UDR)

- Provides network control regarding the service data flow detection, gating, QoS and flow based charging towards the Policy and Charging Enforcement Function (PCEF)
- Receives session and media related information from the Application Function (AF) and informs AF of traffic plane events
- Provisions PCC Rules to the PCEF via the Gx reference point

The CNC Policy interacts with Access and Mobility Management Function (AMF), Session Management Function (SMF), PCRF-Core, and Application Function (AF) to provide policy control rules to the Network Functions (NFs) and also interacts with User Data Repository (UDR) to get the subscriber related information for creating the rules.

The CNC 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

For more information about the CNC Policy supported services, see *Oracle Communications Cloud Native Core Policy User's Guide*.

References

Refer the following documents for more information about Cloud Native Core Policy (CNC Policy):

- Oracle Communications Cloud Native Environment Installation Document
- Oracle Communications Cloud Native Core Policy (CNC Policy) User's Guide

Acronyms and Terminology

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

Table 1-1 Acronyms and Terminology

Acronym	Definition
AF	Application Function
AMF	Access and Mobility Management Function
BSF	Binding Support Function
CHF	Charging Function
CM	Configuration Management
CUSTOMER_REPO	Docker registry address including the port number, if the docker registry has an associated port.

Table 1-1 (Cont.) Acronyms and Terminology

Acronym	Definition
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.
MCC	Mobile Country code
METALLB_ADDRESS_POOL	Address pool which configured on metallb to provide external IPs .
MNC	Mobile Network code
NRF	Network Repository Function
PCF	Policy Control Function
CNPCRF	Cloud Native Policy and Charging Rules Function
SAN	Storage Area Network
SMF	Session Management Function
UDR	Unified Data Repository

2
Installing Cloud Native Core Policy

This chapter describes how to install Cloud Native Core Policy on a cloud native environment.

This chapter contains the following:

Pre-Installation Tasks

In this release, Single Release Bundle provides the following deployment models:

- Converged Deployment
- PCF Deployment
- CNPCRF Deployment

Prior to installing the Cloud Native Core Policy (CNC Policy), perform the following tasks:

Checking the Software Requirements

The following softwares must be installed before installing Cloud Native Core Policy (CNC Policy):

> **Note:**
>
> In this release, Cloud Native Core Policy supports Oracle Communications Cloud Native Environment (OCCNE) 1.5.

Software	Version
Kubernetes	v1.16.7
HELM	v3.0

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

Software	App Version	Notes
alertmanager	0.20.0	Required for Tracing
elasticsearch	7.6.1	Required for Logging
elastic-curator	2.0.2	Required for Logging

Software	App Version	Notes
elastic-exporter	1.1.2	Required for Logging
logs	2..7.0	Required for Logging
kibana	7.6.1	Required for Logging
grafana	5.0.5	Required for Metrics
prometheus	11.0.2	Required for Metrics
prometheus-node-exporter	1.9.0	Required for Metrics
metallb	0.12.0	Required for External
metrics-server	2.10.0	Required for Metric Server
occne-snmp-notifier	0.3.0	Required for Metric Server
tracer	0.13.3	Required for Tracing

Note:

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

```
helm ls
```

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

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

Note:

If you are using Network Repository Function (NRF), install it before proceeding with the Core Policy (CNC Policy) installation.

Checking the Environment Setup

Note:

This section is applicable only when the Cloud Native Core Policy (CNC Policy) is deployed in the environment, other than OCCNE.

Network access

The Kubernetes cluster hosts must have network access to:

- Local helm repository, where the Cloud Native Core Policy (CNC Policy) helm charts are available.
  To check if the Kubernetes cluster hosts have network access to the local helm repository, execute the following command:

  ```
  helm repo update
  ```

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

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

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

  where:

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

  image-name is the docker image name.

  image-tag is the tag the image used for the Cloud Native Core Policy (CNC Policy) pod.

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

Client Machine Requirements

Following are the client machine requirements where the deployment commands executed:

- It should have network access to the helm repository and docker image repository.
- It should have network access to the Kubernetes cluster.
- It should have necessary environment settings to run the `kubectl` and `docker`commands. The environment should have privileges to create namespace in the Kubernetes cluster.

- It should have helm client installed with the **push** plugin. The environment should be configured so that the `helm install` command deploys the software in the Kubernetes cluster.

Server or Space Requirements

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

Secret File Requirement

For enabling HTTPs on Ingress/Egress gateway the following certificates and pem files has to be created before creating secret files for keys:

- ECDSA private Key and CA signed ECDSA Certificate (if initialAlgorithm: ES256)
- RSA private key and CA signed RSA Certificate (if initialAlgorithm: RSA256)
- TrustStore password file
- KeyStore password file
- CA signed ECDSA certificate

Installation Tasks

Downloading Cloud Native Core Policy (CNC Policy) package

To download the Cloud Native Core Policy (CNC Policy) package from MOS:

1. Login to My Oracle Support with your credentials.
2. Select **Patches and Updates** tab to locate the patch.
3. In **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter *Oracle Communications Cloud Native Core - 5G* in **Product** field, select *Oracle Communications Cloud Native Core Policy 1.7.0.0.0* from **Release** drop-down.
5. Click **Search**. The **Patch Advanced Search Results** displays a list of releases.
6. Select the required patch from the search results. The Patch Details window opens.
7. Click **Download**. File Download window appears.
8. Click the **<p********_<release_number>_Tekelec>.zip** file to downlaod the CNC Policy package file.

Pushing the Images to Customer Docker Registry

To Push the images to customer docker resgistry:

1. Untar the Cloud Native Core Policy (CNC Policy) package file to get Cloud Native Core Policy (CNC Policy) docker image tar file.
   ```
   tar -xvzf occnp-pkg-1.7.3.tgz
   ```
   The directory consists of the following:
   - **Cloud Native Core Policy (CNC Policy) Docker Images File:** `occnp-images-1.7.3.tar`
   - **Helm File:**

`occnp-1.7.3.tgz`

- **Readme txt File:**
  `Readme.txt`
- **Checksum for Helm chart tgz file:**
  `occnp-1.7.3.tgz.sha256`
- **Checksum for images' tgz file:**
  `occnp-images-1.7.3.tar.sha256`

2. Load the **occnp-images-1.7.3.tar** file into the Docker system

   ```
   docker load --input /IMAGE_PATH/occnp-images-1.7.3.tar
   ```

3. Verify that the image is loaded correctly by entering this command:

   ```
   docker images
   ```

   Refer Docker Images for more information on docker images available in Cloud Native Core Policy (CNC Policy).

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

   ```
   docker tag occnp/app_info:1.7.3 CUSTOMER_REPO/app_info:1.7.3
   docker push CUSTOMER_REPO/app_info:1.7.3

   docker tag occnp/oc-policy-ds:1.7.3 CUSTOMER_REPO/oc-policy-ds:1.7.3
   docker push CUSTOMER_REPO/oc-policy-ds:1.7.3

   docker tag occnp/ocingress_gateway:1.7.7 CUSTOMER_REPO/
   ocingress_gateway:1.7.7
   docker push CUSTOMER_REPO/ocingress_gateway:1.7.7

   docker tag occnp/oc-pcf-sm:1.7.3 CUSTOMER_REPO/oc-pcf-sm:1.7.3
   docker push CUSTOMER_REPO/oc-pcf-sm:1.7.3

   docker tag occnp/oc-pcf-am:1.7.3 CUSTOMER_REPO/oc-pcf-am:1.7.3
   docker push CUSTOMER_REPO/oc-pcf-am:1.7.3

   docker tag occnp/oc-pcf-ue:1.7.3 CUSTOMER_REPO/oc-pcf-ue:1.7.3
   docker push CUSTOMER_REPO/oc-pcf-ue:1.7.3

   docker tag occnp/oc-audit:1.7.3 CUSTOMER_REPO/oc-audit:1.7.3
   docker push CUSTOMER_REPO/oc-audit:1.7.3

   docker tag occnp/oc-ldap-gateway:1.7.3 CUSTOMER_REPO/oc-ldap-
   gateway:1.7.3
   docker push CUSTOMER_REPO/oc-ldap-gateway:1.7.3

   docker tag occnp/oc-query:1.7.3 CUSTOMER_REPO/oc-query:1.7.3
   docker push CUSTOMER_REPO/oc-query:1.7.3

   docker tag occnp/oc-pre:1.7.3 CUSTOMER_REPO/oc-pre:1.7.3
   ```

```
docker push CUSTOMER_REPO/oc-pre:1.7.3

docker tag occnp/oc-perf-info:1.7.3 CUSTOMER_REPO/oc-perf-info:1.7.3
docker push CUSTOMER_REPO/oc-perf-info:1.7.3

docker tag occnp/oc-diam-gateway:1.7.3 CUSTOMER_REPO/oc-diam-
gateway:1.7.3
docker push CUSTOMER_REPO/oc-diam-gateway:1.7.3

docker tag occnp/oc-diam-connector:1.7.3 CUSTOMER_REPO/oc-diam-
connector:1.7.3
docker push CUSTOMER_REPO/oc-diam-connector:1.7.3

docker tag occnp/oc-pcf-user:1.7.3 CUSTOMER_REPO/oc-pcf-user:1.7.3
docker push CUSTOMER_REPO/oc-pcf-user:1.7.3

docker tag occnp/oc-config-mgmt:1.7.3 CUSTOMER_REPO/oc-config-
mgmt:1.7.3
docker push CUSTOMER_REPO/oc-config-mgmt:1.7.3

docker tag occnp/oc-config-server:1.7.3 CUSTOMER_REPO/oc-config-
server:1.7.3
docker push CUSTOMER_REPO/oc-config-server:1.7.3

docker tag occnp/ocegress_gateway:1.7.7 CUSTOMER_REPO/
ocegress_gateway:1.7.7
docker push CUSTOMER_REPO/ocegress_gateway:1.7.7

docker tag occnp/nrf-client:1.2.5 CUSTOMER_REPO/nrf-client:1.2.5
docker push CUSTOMER_REPO/nrf-client:1.2.5

docker tag occnp/oc-readiness-detector:1.7.3 CUSTOMER_REPO/oc-
readiness-detector:1.7.3
docker push CUSTOMER_REPO/oc-readiness-detector:1.7.3

docker tag occnp/configurationinit:1.2.0 CUSTOMER_REPO/
configurationinit:1.2.0
docker push CUSTOMER_REPO/configurationinit:1.2.0

docker tag occnp/configurationupdate:1.2.0 CUSTOMER_REPO/
configurationupdate:1.2.0
docker push CUSTOMER_REPO/configurationupdate:1.2.0

docker tag occnp/oc-soap-connector:1.7.3 CUSTOMER_REPO/occnp/oc-
soap-connector:1.7.3
docker push CUSTOMER_REPO/occnp/oc-soap-connector:1.7.3

docker tag occnp/oc-pcrf-core:1.7.3 CUSTOMER_REPO/occnp/oc-pcrf-
core:1.7.3
docker push CUSTOMER_REPO/occnp/oc-pcrf-core:1.7.3

docker tag occnp/oc-binding:1.7.3 CUSTOMER_REPO/occnp/oc-
binding:1.7.3
docker push CUSTOMER_REPO/occnp/oc-binding:1.7.3
```

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

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

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

Configuring Database, Creating Users, and Granting Permissions

Cloud Native Core Policy (CNC Policy) microservices use MySQL database to store the configuration and run time data. Following microservices require dedicated MySQL databases created in MySQL data tier.

- Session Management (SM) Service - To store SM and Policy Authorization (PA) session state
- Access and Mobility (AM) Service - To store AM session state
- User Service - To store User information like Policy Data (from UDR) and Policy Counter information (from CHF)
- Config Server - To store configuration data
- Audit Service - To store session state audit data
- PCRF Core service - To store Gx session, Rx Session and User Profile information
- Binding Service - To store context binding information of 4g and 5g subscribers

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

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

Service Name	Default Database Name	Applicable to Deployment
SM Service	occnp_pcf_sm	PCF (if smServiceEnable parameter is enabled in custom yaml file.)
AM Service	occnp_pcf_am	PCF (if amServiceEnable parameter is enabled in custom yaml file.)

Service Name	Default Database Name	Applicable to Deployment
User Service	occnp_pcf_user	PCF (mandatory)
Config Server Service	occnp_config_server	cnPCRF & PCF (mandatory)
Audit Service	occnp_audit_service	PCF (if enabled)
PCRF Core Service	occnp_pcrf_core	cnPCRF (if pcrfCoreEnable parameter is enabled in custom yaml file.)
Binding Service	occnp_binding	cnPCRF & PCF (if bindingEnable parameter is enabled in custom yaml file.)

Apart from the databases created for these microservices, create a database, **occnp_release** (default database name) and it is a mandatory database for PCF and cnPCRF. It will be used to store and manipulate the release versions of all PCF and cnPCRF services on install/upgrade and rollback.
It is recommended to use unique database name when there are multiple instances of CNC Policy deployed in the network and they share the same data tier (MySQL cluster).

It is recommended to create custom unique database name, by simply prefixing the deployment name of the CNC Policy. This way database name uniqueness can be achieved across all deployments. However, you can use any prefix/suffix to create the unique database name. For example, if the OCPCF deployment name is "site1" then the SM Service database can be named as "occnp_pcf_sm_site1".

Refer the Configurable Parameters section for how to override default database names with custom database names.

To configure MYSQL database for the different microservices:

1. Login to the server where the ssh keys are stored and SQL nodes are accessible.
2. Connect to the SQL nodes.
3. Login to the database as a root user.
4. Create database for the different microservices:

```
CREATE DATABASE occnp_audit_service;
CREATE DATABASE occnp_config_server;
CREATE DATABASE occnp_pcf_am;
CREATE DATABASE occnp_pcf_sm;
CREATE DATABASE occnp_pcf_user;
CREATE DATABASE occnp_pcrf_core;
CREATE DATABASE occnp_release;
CREATE DATABASE occnp_binding;
CREATE DATABASE occnp_policyds;
```

5. Create an admin user and grant all the necessary permissions to the user by executing the following command:

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



GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_pcf_sm.* TO 'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_pcf_am.* TO 'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_pcf_user.* TO 'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_config_server.* TO 'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_audit_service.* TO 'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_release.* TO 'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_pcrf_core.* TO 'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
REFERENCES, INDEX ON occnp_binding.* TO 'username'@'%';
FLUSH PRIVILEGES;
```

where:
username is the username and *password* is the password for MYSQL admin user.

For Example: In the below example "occnpadminusr" is used as username, "occnpadminpasswd" is used as password and granting all the permissions to "occnpadminusr". In this example, default database names of micro services are used.

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



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_pcf_user.* 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'@'%';
FLUSH PRIVILEGES;
```

6. Create an application user and grant all the necessary permissions to the user by executing the following command:

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

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE  ON occnp_pcf_sm.* TO
'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_pcf_am.* TO
'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_pcf_user.* TO
'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_config_server.* TO
'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_audit_service.* TO
'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_pcrf_core.* TO
'username'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_binding.* TO
'username'@'%';
```

where:

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

For Example: In the below example "occnpusr" is used as username, "occnppasswd" is used as password and granting the necessary permissions to "occnpusr". In this example, default database names of micro services are used.

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

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE  ON occnp_pcf_sm.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON occnp_pcf_am.* TO
'occnpusr'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON occnp_pcf_user.* 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'@'%';
```

> **Note:**
>
> The database name can be specified in the **envMysqlDatabase** parameter for respective services in the custom-value.yaml file.

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

7. Execute the command, `show grants for username`, to confirm that admin or application user has all the permissions.
   where, *username* is the admin or application user's username.

   For Example,

   ```
   show grants for occnpadminusr
   ```

   ```
   show grants for occnpusr
   ```

8. Exit from database and logout from MYSQL node.
9. Create namespace if already does not exists by entering the command:

   ```
   kubectl create namespace release_namespace
   ```

   where:
   release_namespace is the deployment Cloud Native Core Policy (CNC Policy) namespace used by helm command.
10. Create a kubernetes secret for an admin user and an application user that were created in the step 5 and step 6.
    To create a kubernetes secret for storing database username and password for these users:
    a. Create a yaml file with the application user's username and password with the syntax shown below:

       ```
       apiVersion: v1
       kind: Secret
       metadata:
         name: occnp-db-pass
       type: Opaque
       data:
         mysql-username: b2NjbnB1c3I=
         mysql-password: b2NjbnBwYXNzd2Q=
       ```

    b. Create a yaml file with the admin user's username and password with the syntax shown below:

       ```
       apiVersion: v1
       kind: Secret
       metadata:
         name: occnp-admin-db-pass
       type: Opaque
       data:
       ```

```
mysql-username: b2NjbnBhZG1pbnVzcg==
mysql-password: b2NjbnBhZG1pbnBhc3N3ZA==
```

> **Note:**
>
> 'name' will be used for the **dbCredSecretName** and **privilegedDbCredSecretName** parameters in the CNC Policy custom-values.yaml file.

> **Note:**
>
> The values for **mysql-username** and **mysql-password** should be base64 encoded.

c. Execute the following commands to add the kubernetes secrets in a namespace:

```
kubectl create -f yaml_file_name1 -n release_namespace
kubectl create -f yaml_file_name2 -n release_namespace
```

where:
release_namespace is the deployment namespace used by the helm command.

yaml_file_name1 is a name of the yaml file that is created in step a.

yaml_file_name2 is a name of the yaml file that is created in step b.

Installing CNC Policy Package

To install the Cloud Native Core Policy (CNC Policy) package:

1. Modify the required custom-values.yaml file with the required input parameters. To customize the file, see Customizing Cloud Native Core Policy.

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

> **Note:**
>
> **perf-info** has to be provided proper URL or else it will keep on restarting. [Below is an example of URL for bastion server]:
>
> ```
> perf-info:
>
> configmapPerformance:
>
> prometheus: http://occne-prometheus-server.occne-infra.svc
>
> jaeger=jaeger-agent.occne-infra
> ```

2. > **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.

   a. Install CNC Policy by using Helm2:

   ```
   helm install <helm-chart> --name <release_name> --namespace
   <release_namespace> -f <custom_file> --atomic
         --timeout 600
   ```

   b. Install CNC Policy by using Helm3:

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

   where:
   *helm_chart*is the location of the helm chart extracted from occnp-pkg-1.7.3.0.0.tgz file

   release_name is the release name used by helm command.

   release_namespace is the deployment namespace used by helm command.

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

   For example:

   ```
   helm install /home/cloud-user/occnp-1.7.3.tgz --name occnp --
   namespace occnp -f occnp-1.7.3-custom-values-occnp.yaml --atomic
   ```

   Refer Customizing Cloud Native Core Policy for the sample yaml file.

Parameters in `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 Helm2 and 5m (5 minutes) in Helm3.Specifies the time to wait for any individual kubernetes operation (like Jobs for hooks). Default value is 5m0s. 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 (default: 300s). Here timeout value is not for overall install, but it is for automatic purge on installation failure.

3. You can verify the installation while running the install command by entering this commands:

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

   Press "Ctrl+C" to exit watch mode. We should run the `watch` command on the another terminal.

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

4. Check the installation status by entering this command:

   ```
   helm ls release_name
   ```

   For example:

   ```
   helm ls occnp
   ```

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

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

   For example:

   ```
   kubectl get pod -n occnp
   ```

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

> **Note:**
>
> If the installation is not successful or you do not see the status as Running for all the pods, perform the troubleshooting steps given under Troubleshooting Cloud Native Core Policy (CNC Policy).

3
Customizing Cloud Native Core Policy

This chapter describes how to customize the Cloud Native Core Policy (CNC Policy) deployment in a cloud native environment.

The CNC Policy deployment is customized by overriding the default values of various configurable parameters in the **occnp-1.7.3-custom-values-occnp.yaml**, **occnp-1.7.3-custom-values-pcf.yaml**, and **occnp-1.7.3-custom-values-pcrf.yaml** files.

If you are deploying CNC Policy with Aspen service mesh, you can override the default values of configurable parameters and customize them in the custom_values_occnp-custom-values-pcf-unified-ports.yaml, custom_values_occnp-custom-values-pcrf-unified-ports.yaml, and custom_values_occnp-custom-values-occnp-unified-ports.yaml files.

To customize the custom value files as per the required parameters, perform the following steps:

1. Go to the Oracle Help Center (OHC) Web site: https://docs.oracle.com
2. Navigate to **Industries**->**Communications**->**Cloud Native Core->Release 2.2.1**
3. Click the **CNC Policy Custom Template** link to download the zip file.
4. Unzip the file to get the custom value files. These files are used during installation.
5. Depending on the deployment model, customize the required custom-values.yaml file based on all the parameters described in the Configurable Parameters section.
6. Save the updated custom-values.yaml in the helm chart directory.

Configurable Parameters

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

Global Configurations

These configuration parameters are common for all micro services.

Table 3-1 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
userServiceEnable	Detremines if the user service is enabled or not.	O	True	CNC Policy& PCF	Added in Release 1.7.1	
amServiceEnable	Detremines if the AM service is enabled or not.	O	True	CNC Policy& PCF	Added in Release 1.7.1	
smServiceEnable	Detremines if the SM service is enabled or not.	O	True	CNC Policy& PCF	Added in Release 1.7.1	
ueServiceEnable	Detremines if the UE service is enabled or not.	O	True	CNC Policy& PCF	Added in Release 1.7.1	
nrfClientNfDiscoveryEnable		O	True	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.1	
diamConnectorEnable	Detremines if the diameter connector is enabled or not.	O	True	CNC Policy& PCF	Added in Release 1.7.1	
appinfoServiceEnable	Determines if the app info service is enabled or not.	O	True	CNC Policy& PCF	Added in Release 1.7.1	
performanceServiceEnable	Determines if the performance service is enabled or not.	O	True	CNC Policy& PCF	Added in Release 1.7.1	
pcrfCoreEnable	Detremines if the PCRF core service is enabled or not.	O	True	CNC Policy& cnPCRF	Added in Release 1.7.1	
soapConnectorEnable	Detremines if the soap connector is enabled or not.	O	False	CNC Policy& cnPCRF	Added in Release 1.7.1	
diamGatewayEnable	Detremines if the diameter gateway is enabled or not.	O	True	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.1	

Table 3-1 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
bindingEnable	Detremines if the Binding service is enabled or not.	O	True	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.1	This Parameter value is False for PCF & cnPCRF.
policydsEnable	Detremines if the Data Source service is enabled or not.	O	False	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.1	
ldapGatewayEnable	Detremines if the LDAP Gateway is enabled or not.	O	False	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.1	
nrfClientNfManagementEnable		O	True	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.1	
dockerRegistry	Name of the Docker registry which hosts Cloud Native Core Policy docker images	Yes	Not applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.0	This is a docker registry running in OCCNE bastion server where all OAuth docker images will be loaded. For example, 'occne-bastion:5000'
envMysqlHost	IP address or host name of the MySql server which hosts Cloud Native Core Policy's databases	Yes	Not applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.0	

Table 3-1 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
envMysqlPort	port of the MySql server which hosts Cloud Native Core Policy's databases	Yes	Not applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.0	
envJaegerAgentHost	Hostname or IP address for the jaeger agent	Yes	Not applicable	CNC Policy, PCF, &cnPCRF	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>
dbCredSecretName	Name of the Kubernetes secret object containing Database username and password	Yes	Not applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
privilegedDbCredSecretName	Name of the Kubernetes secret object containing Database username and password for an admin user	Yes	Not applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
releaseDbName	Name of the release database containing release version details	Yes	Not applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	

Table 3-1 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
<service chart name>.image	Docker image name for the service	Yes		CNC Policy, PCF, &cnPCRF	Added in Release 1.0	It is required only when you modify the image name.
<service chart name>.imageTag	Tag the image used for the CNC Policy pod	Yes		CNC Policy, PCF, &cnPCRF	Added in Release 1.0	It is required only when you modify the image tag.
pcfApiRoot	API root of PCF that is used in notification URLs generated by PCF's when sending request to other producer NFs (like NRF, UDR, CHF, etc..)	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. Example: "https://<Helm namespace>-pcf-ingress-gateway:443" pcfApiRoot: "

Core Services

Table 3-2 Customizable Parameters

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	

Table 3-2 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
sm-service.envMysqlDatabase	Name of the database for SM-Service	No	occnp_pcf_sm	CNC Policy & PCF	Added in Release 1.0	
sm-service.envMysqlDatabaseUserService	Name of the database of User Service	No	occnp_pcf_user	CNC Policy & PCF	Added in Release 1.6.x	Same value as "user-service.envMysqlDatabase"
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.
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.
sm-service.defaultBsfApiRoot	Api root of pre-configured BSF	No	Not applicable	CNC Policy & PCF	Added in Release 1.5.x	Required, if PCF uses pre-configured BSF. For Example: "https://bsf.apigateway:8001/"
user-service.envMysqlDatabase	Name of the database for User-Service	No	occnp_pcf_user	CNC Policy & PCF	Added in Release 1.0	

Common Services

Table 3-3 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
cm-service.enableHttps	Flag to enable/ disable HTTPS for cm-service GUI/API	Optional	false	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
config-server.envMysqlDatabase	Name of the database for Config Server service	No	occnp_config_server	CNC Policy & PCF	Added in Release 1.0	
queryservice.envMysqlDatabaseSmService	Specify the database name of SM service	Conditional	occnp_pcf_sm	CNC Policy & PCF	Added in Release 1.6.x	
queryservice.envMysqlDatabaseUserService	Specify the database name of User service	Conditional	occnp_pcf_user	CNC Policy & PCF	Added in Release 1.6.x	Same value as "user-service.envMysqlDatabase"
audit-service.envMysqlDatabase	Name of the database for Audit service	No	occnp_audit_service	CNC Policy & PCF	Added in Release 1.7.1	
perf-info.configmapPerformance.prometheus	Specifies Prometheus server URL	Conditional	http://prometheus-server.prometheus:5802	CNC Policy & PCF	Added in Release 1.0	If no value is specified, PCFs load reported to NRF is always 0.
appinfo.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	Added in Release 1.6.x	If no value is specified, PCF creates a service account at the time of deployment.

Table 3-3 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
appinfo.infraServices	Set this parameter to an empty array if any one of below condition is met: • Deploying on occne 1.4 or lesser 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	
policyds.envMysqlDatabaseConfigServer	Specify the database name of Config Server service		occnp_config_server	CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	
ldap-gateway.serviceAccountName				CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	
pcrf-core.envMysqlDatabase	Name of the database for PCRF-Core	No	occnp_pcrf_core	CNC Policy & cnPCRF	Added in Release 1.0	
binding.envMysqlDatabase	Name of the database for Binding service	No	occnp_binding	CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	
binding.bsfEnabled		No	False	CNC Policy & PCF	Added in Release 1.7.1	

NRF Client

Table 3-4 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.deploymentNrfClientService.envNfNamespace	K8s namespace of PCF	Mandatory	Not Applicable	CNC Policy & PCF	Added in Release 1.6.x	
global.deploymentNrfClientService.nfApiRoot	Api root of PCF	Mandatory	Not Applicable	CNC Policy & PCF	Added in Release 1.6.x	same value as global.pcfApiRoot
nrf-client.configmapApplicationConfig.profile	Contains configuration parameters that goes into nrf-client's config map	Mandatory	Not Applicable	CNC Policy & PCF	Added in Release 1.6.x	Refer below table for config parameters in config-map
nrf-client-nfdiscovery.envJaegerSamplerParam			'1'	CNC Policy & PCF	Added in Release 1.7.1	
nrf-client-nfdiscovery.envJaegerSamplerType			ratelimitimg	CNC Policy & PCF	Added in Release 1.7.1	
nrf-client-nfdiscovery.envJaegerServiceName			pcf-nrf-client-nfdiscovery	CNC Policy & PCF	Added in Release 1.7.1	
nrf-client-nfmanagement.envJaegerSamplerParam			'1'	CNC Policy & PCF	Added in Release 1.7.1.0	
nrf-client-nfmanagement.envJaegerSamplerType			ratelimiting	CNC Policy & PCF	Added in Release 1.7.1	
nrf-client-nfmanagement.envJaegerServiceName			pcf-nrf-client-nfmanagement	CNC Policy & PCF	Added in Release 1.7.1	

Config parameters in Config-map

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
primaryNrfApiRoot	Primary NRF API root <http scheme>://<Hostname/IP>:<Port>	valid api root	CNC Policy & PCF	Added in Release 1.6.x	For Example: http://nrf1-api-gateway.svc:80
SecondaryNrfApiRoot	secondary NRF API root <http scheme>://<Hostname/IP>:<Port>	valid api root	CNC Policy & PCF	Added in Release 1.6.x	For Example: http://nrf2-api-gateway.svc:80
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
nrfClientType	This should be set to PCF	PCF	CNC Policy & PCF	Added in Release 1.6.x	
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.
appProfiles	NfProfile of PCF to be registered with NRF	Valid NF Profile	CNC Policy & PCF	Added in Release 1.6.x	
enableF3	Support for 29.510 Release 15.3	true/false	CNC Policy & PCF	Added in Release 1.6.x	
enableF5	Support for 29.510 Release 15.5	true/false	CNC Policy & PCF	Added in Release 1.6.x	
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)
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)
enableSubscriptionAutoRenewal	Enable Renewal of Subscriptions automatically	true/false	CNC Policy & PCF	Added in Release 1.6.x	

Parameter	Description	Allowed Values	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
acceptAdditionalAttributes	Enable additionalAttributes as part of 29.510 Release 15.5	true/false	CNC Policy & PCF	Added in Release 1.6.x	
supportedDataSetId		POLICY	CNC Policy & PCF	Added in Release 1.7.1	

Diameter

Table 3-5 Customizable Parameters

Parameter	Description	Mandatory 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
diam-connector.envDiameterIdentity	Diameter Host of PCF	Yes	Not applicable	CNC Policy & PCF	Added in Release 1.6.x	example: ocpcf
diam-gateway.envGatewayMode	Diameter Gateway mode	Yes		CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	For CNC Policy,the value is "converged". For PCF,the value is "PCF". For cnPCRF,the value is "cnPCRF".
diam-gateway.envGatewayDeploymentType	Diameter Gateway deployment type (applicable only when mode is converged)	Yes		CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	For CNC Policy,the value is "CONVERGED". For PCF,the value is "PCF". For cnPCRF,the value is "cnPCRF".
diam-gateway.envDiameterRealm	Diameter Realm of PCF diameter gateway	Yes	Not applicable	CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	example: oracle.com

Table 3-5 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
diam-gateway.envDiameterIdentity	Diameter Host of PCF diameter gateway	Yes	Not applicable	CNC Policy, PCF, & cnPCRF	Added in Release 1.7.1	example: oc-diam-gateway

Ingress Gateway Service

Table 3-6 Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
global.publicHttpSignalingPort	HTTP/2.0 Port of ingress gateway	No	80	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	
global.publicHttpsSignallingPort	HTTPS/2.0 Port of ingress gateway	No	443	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	
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.enabled	Determines if ingress gateway is enabled or not.		True	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	

Table 3-6 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.serviceMeshCheck	Enable this parameter if load balancing is handled by Service Mesh	No	False	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	
ingress-gateway.jaegerTracingEnabled		No	False	CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
ingress-gateway.openTracing.jaeger.udpSender.host				CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
ingress-gateway.openTracing.jaeger.udpSender.port				CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
ingress-gateway.openTracing.jaeger.probabilisticSampler				CNC Policy, PCF, &cnPCRF	Added in Release 1.6.x	
ingress-gateway.oauthValidatorEnabled	Enable or disable OAuth Validator	Yes	False	CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.nfInstanceId	NF Instance Id of service producer	No	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	No	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	No		CNC Policy & PCF	Added in Release 1.5.x	

Table 3-6 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.nrfPublicKeyKubeNamespace	Namespace of the NRF public key secret	No		CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.validationType	Possible values are: • strict • relaxed strict- If incoming request does not contain "Authorization" (Access Token) header, the request is rejected. relaxed- relaxed means that if Incoming request contains "Authorization" header, it is validated. If Incoming request does not contain "Authorization" header, validation is ignored.	No		CNC Policy & PCF	Added in Release 1.6.x	
ingress-gateway.producerPlmnMNC	MNC of the service producer	No		CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.producerPlmnMCC	MCC of the service producer	No		CNC Policy & PCF	Added in Release 1.5.x	
ingress-gateway.enableIncomingHttp	To enable http (INSECURE) for ingress traffic	No	False	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	
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 privatekey secret	No	Not Applicable	CNC Policy, PCF, &cnPCRF	Added in Release 1.5.x	required if enableIncomingHttps is true

Table 3-6 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingress-gateway.service.ssl.privateKey.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.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.privateKey.ecdsa.fileName	ecdsa private key file name	No	Not Applicable		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.certificate.ecdsa.fileName	ecdsa private key file name	No	Not Applicable		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
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

Table 3-6 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
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
ingressServer.keepAlive.enabled		No	false		Added in Release 1.7.3	
ingressServer.keepAlive.idealTime		No	180 (in seconds)		Added in Release 1.7.3	
ingressServer.keepAlive.count		No	9		Added in Release 1.7.3	

Table 3-6 (Cont.) Customizable Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/ Updated in Release	Notes
ingressServer.keepAlive.interval		No	60 (in seconds)		Added in Release 1.7.3	
global.configServerPort		No	5807	CNC Policy, PCF, &cnPCRF	Added in Release 1.7.3	

Egress Gateway Service

Table 3-7 Customization Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Modified in Release	Notes
egress-gateway.enabled	Determines if egress gateway is enabled or not.		True	CNC Policy, PCF, & cnPCRF	Added in Release 1.5.x	
egress-gateway.jaegerTracingEnabled		No	False	CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.openTracing.jaeger.udpSender.host	udpsender host			CNC Policy& PCF	Added in Release 1.7.1	
egress-gateway.openTracing.jaeger.udpSender.port	udpsender port			CNC Policy& PCF	Added in Release 1.7.1	
egress-gateway.openTracing.jaeger.probabilisticSampler				CNC Policy& PCF	Added in Release 1.7.1	
egress-gateway.oauthClientEnabled	OAuth Validator Enabled	No	false	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.nrfAuthority	NRF's ${HOSTNAME}:{PORT}	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	Modify the parameter with actual value, if oAuth is enabled.

Table 3-7 (Cont.) Customization Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Modified in Release	Notes
egress-gateway.nfInstanceId	NF InstanceId of Producer	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	Modify the parameter with actual value, if OAuth is enabled.
egress-gateway.consumerPlmnMNC	MNC of service Consumer	No		CNC Policy& PCF	Added in Release 1.5.x	Modify the parameter with actual value, if OAuth is enabled.
egress-gateway.consumerPlmnMCC	MCC of service Consumer	No		CNC Policy& PCF	Added in Release 1.5.x	Modify the parameter with actual value, if OAuth is enabled.
egress-gateway.enableOutgoingHttps	Enabling it for outgoing https request	No		CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.egressGwCertReloadEnabled		No		CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.egressGwCertReloadPath		No		CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.privateKey.k8SecretName	Name of the privatekey secret	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
egress-gateway.service.ssl.privateKey.k8NameSpace	Namespace of privatekey	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	

Table 3-7 (Cont.) Customization Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Modified in Release	Notes
egress-gateway.service.ssl.certificate.k8NameSpace	Namespace of privatekey	No	Not Applicable	CNC Policy& PCF	Added in Release 1.5.x	
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	

Table 3-7 (Cont.) Customization Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Modified in Release	Notes
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	
egress-gateway.scpIntegrationEnabled	Change this to false when scp integration is not required	No	false	CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.scp.scpRerouteEnabled	Set this flag to true if re-routing to multiple SCP instances is to be enabled. globalretry can be enabled only when scpRerouteEnabled flag is set to true.	No	false	CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.globalretry.enabled	globalretry can be enabled only when scpRerouteEnabled flag is set to true. And, it is applied only when no "retries" is specified under routesConfig.	O	false	CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.globalretry.retries				CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.scp.instances.http.host	SCP HTTP IP/ FQDN	No	Not Applicable	CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.scp.instances.http.Port	SCP HTTP PORT	No	80	CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.scp.instances.http.ApiPrefix	Change this value to corresponding prefix "/" is not expected to be provided along. Applicable only for SCP with TLS enabled.	No	/	CNC Policy& PCF	Added in Release 1.6.x	

Table 3-7 (Cont.) Customization Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Modified in Release	Notes
egress-gateway.scp.scpDefaultScheme	Default scheme applicable when 3gpp-sbi-target-apiroot header is missing	No	https	CNC Policy& PCF	Added in Release 1.6.x	
egress-gateway.K8ServiceCheck	Enable this if loadbalancing is to be done by egress instead of K8s	No	false	CNC Policy& PCF	Added in Release 1.5.x	
httpsScpOnly	This is global parameter which will be taken into consideration if route (under routeConfig section) based httpsScpOnly parameter is not available. If set to true, select SCP instances for https list only. If set to false, run existing logic as per provided scheme.	No	false	CNC Policy& PCF	Added in Release 1.7.3	Please note double quotes to be enclosed for values of httpScpOnly.
httpRuriOnly	This is global parameter which will be taken into consideration if route (under routeConfig section) based httpRuriOnly parameter is not available. If set to true, change scheme of RURI to http. If set to false, don't change the scheme.	No	false	CNC Policy& PCF	Added in Release 1.7.3	Please notedouble quotes to be enclosed for values of httpsScpOnly.

Table 3-7 (Cont.) Customization Parameters

Parameter	Description	Mandatory Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Modified in Release	Notes
routesConfig[0].httpRuriOnly	If set to true, change Scheme of RURI to http. If set to false, don't change the scheme.	No	false	CNC Policy& PCF	Added in Release 1.7.3	Please note double quotes to be enclosed for values of httpsRuriOnly. If httpsRuriOnly under route is not present globally available value will be considered.
routesConfig[0].httpsScpOnly	If set to true, select SCP instances for https list only. If set to false, run existing logic as per provided scheme.	No	false	CNC Policy& PCF	Added in Release 1.7.3	Please note double quotes to be enclosed for values of httpsScpOnly. If httpsScpOnly under route is not present globally available value will be considered.

Additional Configurable Parameters for Aspen mesh

This section describes the customizatons that you can make in custom_values_occnp-custom-values-pcf-unified-ports.yaml, custom_values_occnp-custom-values-pcrf-unified-ports.yaml, and custom_values_occnp-custom-values-occnp-unified-ports.yaml files to integrate Aspen service mesh with Oracle Communications Cloud Native Core Policy.

> **Important:**
>
> Users may use custom values file from CNC Policy 1.7.0 to install CNC Policy with Aspen service mesh.

- **Unified signaling ports**: To override the default port numbers, used by containers and services, and customize them as per your requirements, you can configure the following configurable parameters in custom values file:

Table 3-8 Customizable service ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
servicePorts.pcfAmServiceHttp	HTTP signaling port for AM service.	Optional	5904	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcfAmServiceHttp	HTTP signaling port for AM service.	Optional	5905	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.appInfoHttp	HTTP signaling port for app info .	Optional	5906	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcAppInfoHttp`
servicePorts.auditServiceHttp	HTTP signaling port for audit service.	Optional	5807	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.bindingHttp	HTTP signaling port for binding service.	Optional	8080	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.bindingHttps	HTTPS signaling port for binding service.	Optional	8443	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.cmServiceHttp	HTTP signaling port for CM service.	Optional	5808	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.configServerHttp	HTTP signaling port for config server.	Optional	5807	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	Same value as `svcConfigServerHttp`
servicePorts.pcfDiamConnectorHttp	HTTP signaling port for PCF Diameter connector.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcfDiamConnectorDiameter	Port for PCF Diameter connector.	Optional	3868	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.ldapGatewayHttp	HTTP signaling port for LDAP Gateway.	Optional	8084	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.ldapGatewayHttps	HTTPS signaling port for LDAP Gateway.	Optional	8443	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	

Table 3-8 (Cont.) Customizable service ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
servicePorts.pcfDiamGatewayHttp	HTTP signaling port for PCF Diameter gateway.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcfDiamGatewayDiameter	Port for PCF Diameter gateway.	Optional	3868	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcrfCoreDiameter	Port for PCRF Core Diameter.	Optional	3868	CNCPolicy & cnPCRF	Added in Release 1.7.3	
servicePorts.pcrfCoreHttp	HTTP signaling port for PCRF core service.	Optional	9080	CNCPolicy & cnPCRF	Added in Release 1.7.3	
servicePorts.pcrfDiamGatewayHttp	HTTP signaling port for PCRF Diameter Gateway.	Optional	8080	CNCPolicy & cnPCRF	Added in Release 1.7.3	
servicePorts.pcrfDiamGatewayDiameter	Port for PCRF Diameter connector.	Optional	3868	CNCPolicy & cnPCRF	Added in Release 1.7.3	
servicePorts.perfInfoHttp	HTTP signaling port for perf info.	Optional	5905	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcPerfInfoHttp`
servicePorts.policydsHttp	HTTP signaling port for policyds.	Optional	8080	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.preServiceHttp	HTTP signaling port for pre service.	Optional	5806	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.preTestHttp	HTTP signaling port for pre test.	Optional	5806	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.queryServiceHttp	HTTP signaling port for queryservice.	Optional	5805	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
servicePorts.pcfSmServiceHttp	HTTP signaling port for SM service.	Optional	5809	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcfSmServiceHttps	HTTPS signaling port for SM service.	Optional	5805	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.soapConnectorHttp	HTTP signaling port for Soap connector.	Optional	8082	CNCPolicy & cnPCRF	Added in Release 1.7.3	

Table 3-8 (Cont.) Customizable service ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
servicePorts.pcfUeServiceHttp	HTTP signaling port for UE service.	Optional	5809	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcfUeServiceHttps	HTTPS signaling port for UE service.	Optional	5805	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcfUserServiceHttp	HTTP signaling port for User service.	Optional	5808	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.pcfUserServiceHttps	HTTPS signaling port for User service.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.udrConnectorHttp	HTTP signaling port for UDR Connector.	Optional	5808	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.udrConnectorHttps	HTTPS signaling port for UDR Connector.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.chfConnectorHttp	HTTP signaling port for CHF Connector.	Optional	5808	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.chfConnectorHttps	HTTPS signaling port for CHF Connector.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
servicePorts.egressGatewayHttp	HTTP signaling port for Egress Gateway.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcEgressGatewayHttp`
servicePorts.nrfClientNfDiscoveryHttp	HTTP signaling port for NRF client discovery service.	Optional	5910	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfDiscoveryHttp`

Table 3-8 (Cont.) Customizable service ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
servicePorts.nrfClientNfManagementHttp	HTTP signaling port for NRF client management service.	Optional	5910	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfManagementHttp`
servicePorts.nrfClientNfDiscoveryHttps	HTTPS signaling port for NRF client discovery service.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfDiscoveryHttps`
servicePorts.nrfClientNfManagementHttps	HTTPS signaling port for NRF client management service.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `svcNrfClientNfManagementHttps`

Table 3-9 Customizable container ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
containerPorts.monitoringHttp	HTTP signaling port for monitoring.	Optional	9000	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	Same value as `containerMonitoringHttp`
containerPorts.pcfAmServiceHttp	HTTP signaling port for AM service.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	

Table 3-9 (Cont.) Customizable container ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
containerPorts.pcfAmServiceHttps	HTTPS signaling port for AM service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.appInfoHttp	HTTP signaling port for app info.	Optional	5906	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.auditServiceHttp	HTTP signaling port for Auditservice.	Optional	8081	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.bindingHttp	HTTP signaling port for binding service.	Optional	8080	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.bindingHttps	HTTPS signaling port for binding service.	Optional	8443	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.cmServiceHttp	HTTP signaling port for CMservice.	Optional	5807	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.configServerHttp	HTTP signaling port for config server.	Optional	8001	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.pcfDiamConnectorHttp	HTTP signaling port for Diameter Connector.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.pcfDiamConnectorDiameter	PCF diameter connector.	Optional	3868	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.ldapGatewayHttp	HTTP signaling port for IDAP Gateway.	Optional	8084	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.pcfDiamGatewayHttp	HTTP signaling port for Diameter Gateway.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.pcfDiamGatewayDiameter	PCF diameter gateway.	Optional	3868	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.pcrfCoreDiameter	PCRF core diameter.	Optional	3868	CNCPolicy & cnPCRF	Added in Release 1.7.3	
containerPorts.pcrfCoreHttp	HTTP signaling port for PCRF Core service.	Optional	9080	CNCPolicy & cnPCRF	Added in Release 1.7.3	

Table 3-9 (Cont.) Customizable container ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
containerPorts.pcrfDiamGatewayHttp	HTTP signaling port for PCRF Diameter Gateway.	Optional	8080	CNCPolicy & cnPCRF	Added in Release 1.7.3	
containerPorts.pcrfDiamGatewayDiameter	PCRF diameter gateway.	Optional	3868	CNCPolicy & cnPCRF	Added in Release 1.7.3	
containerPorts.perfInfoHttp	HTTP signaling port for perf-info.	Optional	5905	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.policydsHttp	HTTP signaling port for policyds.	Optional	8080	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.preServiceHttp	HTTP signaling port for pre service.	Optional	5806	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.preTestHttp	HTTP signaling port for pre test.	Optional	5806	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.queryServiceHttp	HTTP signaling port for queryservice.	Optional	8081	CNCPolicy , PCF, &cnPCRF	Added in Release 1.7.3	
containerPorts.pcfSmServiceHttp	HTTP signaling port for SM service.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.pcfSmServiceHttps	HTTPS signaling port for SM service.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.soapConnectorHttp	HTTP signaling port for soap connector.	Optional	8082	CNCPolicy & cnPCRF	Added in Release 1.7.3	
containerPorts.pcfUeServiceHttp	HTTP signaling port for UE service.	Optional	8082	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.pcfUeServiceHttps	HTTPS signaling port for UE service.	Optional	8081	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.pcfUserServiceHttp	HTTP signaling port for User service.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.pcfUserServiceHttps	HTTPS signaling port for User service.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.udrConnectorHttp	HTTP signaling port for UDR Connector.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	

Table 3-9 (Cont.) Customizable container ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
containerPorts.udrConnectorHttps	HTTPS signaling port for UDR Connector.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.chfConnectorHttp	HTTP signaling port for CHF connector.	Optional	8080	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.chfConnectorHttps	HTTPS signaling port for CHF connector.	Optional	8443	CNCPolicy & PCF	Added in Release 1.7.3	
containerPorts.nrfClientNfDiscoveryHttp	HTTP signaling port for NRF client discovery.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfDiscoveryHttp`
containerPorts.nrfClientNfManagementHttp	HTTP signaling port for NRF client management.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfManagementHttp`
containerPorts.nrfClientNfDiscoveryHttps	HTTPS signaling port for NRF client discovery.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfDiscoveryHttps`

Table 3-9 (Cont.) Customizable container ports

Parameter	Description	Mandatory/ Optional Parameter	Default Value	Applicable to Deployment	Added/ Deprecated/Updated in Release	Notes
containerPorts.nrfClientNfManagementHttps	HTTPS signaling port for NRF client management.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerNrfClientNfManagementHttps`
containerPorts.ingressGatewayHttp	HTTP signaling port for Ingress Gateway.	Optional	8000	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerIngressGatewayHttp`
containerPorts.ingressGatewayHttps	HTTPS signaling port for Ingress Gateway.	Optional	9443	CNCPolicy & PCF	Added in Release 1.7.3	Same value as `containerIngressGatewayHttps`

> **Note:**
>
> After you install CNC policy, you can see that all the services of type ClusterIP exposes HTTP on port 8000 and HTTPS on port 9443.

- **Annotation to support OSO**: To deploy CNC Policy with OSO, you must add the following annotation to the custom extension under global section of custom values file:

```
global:
  customExtension:
    lbDeployments:
      annotations:
          oracle.com/cnc: "true"

    nonlbDeployments:
```

```
annotations:
    oracle.com/cnc: "true"
```

> **Note:**
>
> After helm install is complete, all the nodes will have the above mentioned annotation.

- **Custom container name**: You can 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 custom values 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:
```

- **Custom service account**: You can use a custom service account for all services by adding it to global section in the custom values file:

```
global:
  serviceAccountName: ocpcfsaccount
```

> **Note:**
>
> You can create the service account and roles before the installation as well.

- **Disable init containers**: Init containers do not work when the namespace has aspen service mTLS enabled. To disable init containers, set the value for `initContainerEnable` to false in custom values file.

```
global:
  initContainerEnable: false
```

- **PERMISSIVE rule**: To set Permissive rule for Diameter Gateway and Ingress Gateway Service, set the following flags to true in custom value file:

```
global:
  istioIngressTlsSupport:
      diamGateway: true


global:
  istioIngressTlsSupport:
    ingressGateway: true
```

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.

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. Replace line 34 text, **type: NodePort** with **type: LoadBalancer**.
   Following is the sample content after replacing the line 29:

   ```
   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: LoadBalancer
35 status:
36   loadBalancer: {}
```

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

```
NAME                          TYPE
CLUSTER-IP      EXTERNAL-IP
PORT(S)                                                      AGE
<PCRF_NAME>-diam-gateway-service       LoadBalancer
10.xxx.xx.xx    10.xxx.xxx.xx   3868:32592/TCP,8080:31301/TCP
4d
```

Updating cm-service

Follow the same process to update svc yaml for *PCRF_NAME* -pcrf-cm-service.

5 Uninstalling Cloud Native Core Policy (CNC Policy)

When you uninstall a Helm chart from your Cloud Native Core Policy (CNC Policy) deployment, it removes only the Kubernetes objects that it created during installation.

To uninstall, enter this command:

```
helm delete release_name
```

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

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

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

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

For example, to completely remove a release named "occnp",enter this command:

```
helm delete --purge occnp
```

Deleting Kubernetes Namespace

To delete kubernetes namespace, enter this command:

```
kubectl delete namespace release_namespace
```

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

For example, to delete a kubernetes namespace named "occnp", enter this command:

```
kubectl delete namespace occnp
```

Cleaning Up Database

To clean up database for the different microservices:

```
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_pcf_user;
DROP DATABASE IF EXISTS occnp_pcrf_core;
```

```
DROP DATABASE IF EXISTS occnp_release;
DROP DATABASE IF EXISTS occnp_binding;
```

6

Migrating Data from Release 1.6.x to Release 1.7.x

There can be PCF/cnPCRF deployments that are already configured and you want to migrate the configuration from PCF/cnPCRF 1.6.x to CNC Policy 1.7.x.

In the CNC Policy release 1.7.x, you can use Clod Native Core Console and REST APIs to bulk import the data that is exported from 1.6.x release stetups, these REST APIs performs conversion and migration to sort out the incompatibility between the releases.

For more information on Bulk Import/Export, See Bulk Export/Bulk Import section in *Oracle Communications Cloud Native Core Policy User's Guide*.

7 Troubleshooting Cloud Native Core Policy (CNC Policy)

This section provides information to troubleshoot the common error which can be encountered during the installation and upgrade of Cloud Native Core Policy (CNC Policy).

If `helm install` command Fails

This section covers the reasons and troubleshooting procedures if the `helm install` command fails.

Reasons for `helm install` failure:

- **Chart syntax issue [This issue could be shown in the few seconds]**
  Please resolve the chart specific things and rerun the `helm install` command, because in this case, no hooks should have begun.
- **Most possible reason [TIMEOUT]**
  If any job stuck in a pending/error state and not able to execute, it will result in the timeout after 5 minutes. As default timeout for helm command is "5 minutes". In this case, we have to follow the below steps to troubleshoot.
- **`helm install` command failed in case of duplicated chart**

  ```
  helm install /home/cloud-user/pcf_1.6.1/sprint3.1/ocpcf-1.6.1-sprint.3.1.tgz --name ocpcf2 --namespace ocpcf2 -f <custom-value-file>
  ```

  Error: release ocpcf2 failed: configmaps "perfinfo-config-ocpcf2" already exists

  Here, configmap 'perfinfo-config-ocpcf2' exists multiple times, while creating Kubernetes objects after pre-upgrade hooks, this will be failed. In this case also please go through the below troubleshooting steps.
  Troubleshooting steps:

  1. Check from describe/logs of failure pods and fix them accordingly. You need to verify what went wrong on the installation of the CNC Policy by checking the below points:
     For the PODs which were not started, run the following command to check the failed pods:

     ```
     kubectl describe pod <pod-name>  -n <release-namespace>
     ```

     For the PODs which were started but failed to come into "READY"state, run the following command to check the failed pods:

     ```
     kubectl describe logs <pod-name>  -n <release-namespace>
     ```

2. Execute the below command to get kubernetes objects:

   ```
   kubectl get all -n <release_namespace>
   ```

   This gives a detailed overview of which objects are stuck or in a failed state.
3. Execute the below command to delete all kubernetes objects:

   ```
   kubectl delete all --all -n <release_namespace>
   ```
4. Execute the below command to delete all current configmaps:

   ```
   kubectl delete cm --all -n <release-namespace>
   ```
5. Execute the below command to cleanup the databases created by the `helm install` command and create the database again:

   ```
   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_pcf_user;
   DROP DATABASE IF EXISTS occnp_pcrf_core;
   DROP DATABASE IF EXISTS occnp_release;
   DROP DATABASE IF EXISTS occnp_binding;
   CREATE DATABASE IF NOT EXISTS occnp_audit_service;
   CREATE DATABASE IF NOT EXISTS occnp_config_server;
   CREATE DATABASE IF NOT EXISTS occnp_pcf_am;
   CREATE DATABASE IF NOT EXISTS occnp_pcf_sm;
   CREATE DATABASE IF NOT EXISTS occnp_pcf_user;
   CREATE DATABASE IF NOT EXISTS occnp_pcrf_core;
   CREATE DATABASE IF NOT EXISTS occnp_release;
   CREATE DATABASE IF NOT EXISTS occnp_binding;
   ```
6. Execute the below command :

   ```
   helm ls --all
   ```

   If this is in a failed state, please purge the namespace using the command

   ```
   helm delete --purge <release_namespace>
   ```

> **Note:**
>
> If the execution of this command is taking more time, run the below command parallelly in another session to clear all the delete jobs.
>
> ```
> while true; do kubectl delete jobs --all -n
> <release_namespace>; sleep 5;done
> ```

Monitor the below command:

```
helm delete --purge <release_namespace>
```

Once that is succeeded, press "ctrl+c" to stop the above script.

7. After the database cleanup and creation of the database again, run the `helm install` command.

You can use **Data Collector** tool to fetch Network Function (NF) specific logs, metrics, traces, alerts from production environment integrated with Elastic search and Prometheus. See *Cloud Native Core NF Data Collector User's Guide* for more information.

A
Docker Images

Cloud Native Core Policy (CNC Policy) deployment package includes ready-to-use docker images and Helm charts to help you orchestrate containers in Kubernetes.

You can use the Docker images and Helm chart to help you deploy and manage Pods of Cloud Native Core Policy (CNC Policy) product services in Kubernetes. Communication between Pods of services of Cloud Native Core Policy (CNC Policy) products are preconfigured in the Helm charts.

Table A-1 lists the docker images for Cloud Native Core Policy (CNC Policy).

Table A-1 Docker Images for Cloud Native Core Policy (CNC Policy)

Service Name	Docker Image Name
AM Service	oc-pcf-am
Application Info Service	app_info
Binding Service	oc-binding
CM Service	oc-config-mgmt
Config Server Service	oc-config-server
Diameter Connector	oc-diam-connector
Diameter Gateway	oc-diam-gateway
Egress Gateway	ocegress_gateway
Ingress Gateway	ocingress_gateway
Ingress/Egress Gateway init configuration	configurationinit
Ingress/Egress Gateway update configuration	configurationupdate
LDAP Gateway Service	oc-ldap-gateway
Nrf Client Service	nrf-client
PCRF Core Service	oc-pcrf-core
Performance Monitoring Service	oc-perf-info
PolicyDS Service	oc-policy-ds
Policy Runtime Service	oc-pre
Query Service	oc-query
Readiness check	oc-readiness-detector
Session State Audit	oc-audit
SM Service	oc-pcf-sm
Soap Connector	oc-soap-connector
UE Service	oc-pcf-ue
User Service	oc-pcf-user

B

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).

C

Integrating Aspen with CNC Policy

Perform the following steps to integrate Aspen service mesh with CNC Policy:

1. To create a privileged pod security policy for PCF namespace pcfaspen, create a YAML file (`pcf.priv.yaml`) using the following sample code:

```
# permit access to all service accounts in the namespace.
apiVersion:rbac.authorization.k8s.io/v1
kind:RoleBinding
metadata:
  name:"psp:pcfaspen:cs-restricted"
  namespace:"pcfaspen"
roleRef:
  kind:ClusterRole
  apiGroup:rbac.authorization.k8s.io
  name:"psp:privileged"
subjects:- kind:Group
  apiGroup:rbac.authorization.k8s.io
  name:"system:serviceaccounts"
```

2. Add the destination-rule for mysql, prometheus and nf1stub services to let pcfaspen namespace be enabled with ISTIO-Injection. To do so, create a YAML file (`aspendestinationrule.yaml`) using the following sample code:

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: mysql-mysql
  namespace: pcfaspen
spec:
  host: "mysql.mysql.mysqlaspen.svc.cluster.local"
  trafficPolicy:
    tls:
      mode: DISABLE

---

apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: prometheus
  namespace: pcfaspen
spec:
  host: "prometheus-server.infra.svc.cluster.local"
  trafficPolicy:
    tls:
      mode: DISABLE
```

```
---

apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: nf1stub
  namespace: pcfaspen
spec:
  host: "nf1stub.ocats.svc.cluster.local"
  trafficPolicy:
    tls:
      mode: DISABLE
```

Apply the configuration in `aspendestinationrule.yaml` file by entering following command:

```
kubectl apply -f aspendestinationrule.yaml
```

> **Note:**
>
> You may ignore these destination roles if you are deploying Aspen without mTLS.

Then, run the following command in every MySQL node:

```
mysqladmin -h 127.0.0.1 -u "username" -p "password" flush-hosts
```

3. Create namespace `pcfaspen` by running the following command:

```
kubectl create ns pcfaspen
kubectl label --overwrite namespace pcfaspen istio-injection=enabled
```

4. Create secret for privileged and application database user by running the following commands:

```
kubectl create -f priv-secret.yaml -n pcfaspen;
kubectl create -f secret.yaml -n pcfaspen;
```

5. Create privileged pod security policy for namespace created in step 3.

```
kubectl create -f pcf.priv.yaml -n pcfaspen;
```

6. Then, perform steps 2-4 under Installation Tasks to install CNC Policy package.
7. Set the `initContainerEnable` flag to false in the custom value file of occnp.

```
global:
        initContainerEnable: false
```

See Customizing Cloud Native Core Policy for detailed instructions on how to customize the custom value file of occnp.

8. Run the following helm command:

```
helm3 install pcfaspen occnp/  -n pcfaspen -f occnp-1.7.3-custom-
values-occnp.yaml
```

9. Add policy to make cm-service enable the traffic for both encrypted as well as clear-text. To do so, create a YAML file (`aspenpolicy.yaml`) using the following sample code:

```
apiVersion: "authentication.istio.io/v1alpha1"
kind: Policy
metadata:
  name: cmservice
  namespace: pcfaspen
spec:
  targets:
  - name: pcfaspen-occnp-config-mgmt
  peers:
  - mtls:
      mode: PERMISSIVE
```

   Apply the configuration in `aspenpolicy.yaml` file by entering following command:

```
kubectl apply -f aspenpolicy.yaml
```

10. Add service entry for stub service to avoid accessing the pod ID directly. To do so, create a YAML file (`AspenServiceEntry.yaml`) using the following sample code:

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: ats-stubaccess
  namespace: ocats
spec:
  addresses:
  - 10.233.67.12
  hosts:
  - nf1stub.ocats.svc.cluster.local
  location: MESH_EXTERNAL
  ports:
  - name: http
    number: 8080
    protocol: HTTP
  resolution: NONE
```

   Apply the configuration in `AspenServiceEntry.yaml` file by entering following command:

```
kubectl apply -f AspenServiceEntry.yaml
```

Verify Aspen service mesh

After successfully installing Aspen mesh, make sure to verify:

- All pods contain sidecar proxy container by running the following command:

  ```
  kubectl describe pod <pod-name> -n <namespace>
  ```

  > **Note:**
  >
  > Perform this step for all pods.

- Internal traffic flowing between PCF services under the PCF namespace.

  > **Note:**
  >
  > To perform this step, you must sign in to Aspen user interface.

Disabling Aspen service mesh

To disable Aspen service mesh, perform the following steps:

1. Run `kubectl label` command by removing last enabled value and keeping empty label for PCF namespace:

   ```
   kubectl label --overwrite namespace <pcf-namespace> istio-injection=
   ```

2. Restart all PCF pods. The new pods will contain only service containers.

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

D

Upgrading CNC Policy (1.7.x to 1.7.3)

This appendix describes the procedure to upgrade CNC Policy from 1.7.x to 1.7.3.

> **Note:**
>
> Take a backup of all the configurations before upgrade and no manual configuration should be performed during upgrade.

You can select the deployment model by selecting the different custom yaml file in release site, for example:

Released Custom yaml File	Purpose
occnp-1.7.3-custom-values-occnp.yaml	This is the custom yaml file for converged installation.
occnp-1.7.3-custom-values-pcf.yaml	This is the custom yaml file for PCF installation.
occnp-1.7.3-custom-values-pcrf.yaml	This is the custom yaml file for cnPCRF installation.

You can download the required custom yaml file from OHC.

Downloading Cloud Native Core Policy (CNC Policy) package

To download the Cloud Native Core Policy (CNC Policy) package from MOS:

1. Login to My Oracle Support with your credentials.
2. Select **Patches and Updates** tab to locate the patch.
3. In **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter *Oracle Communications Cloud Native Core - 5G* in **Product** field, select *Oracle Communications Cloud Native Core Policy 1.7.0.0.0* from **Release** drop-down.
5. Click **Search**. The **Patch Advanced Search Results** displays a list of releases.
6. Select the required patch from the search results. The Patch Details window opens.
7. Click **Download**. File Download window appears.
8. Click the **<p********_<release_number>_Tekelec>.zip** file to downlaod the CNC Policy package file.

Pushing the Images to Customer Docker Registry

To Push the images to customer docker resgistry:

1. Untar the Cloud Native Core Policy (CNC Policy) package file to get Cloud Native Core Policy (CNC Policy) docker image tar file.

```
tar -xvzf occnp-pkg-1.7.3.tgz
```

The directory consists of the following:

- **Cloud Native Core Policy (CNC Policy) Docker Images File:** `occnp-images-1.7.3.tar`
- **Helm File:** `occnp-1.7.3.tgz`
- **Readme txt File:** `Readme.txt`
- **Checksum for Helm chart tgz file:** `occnp-1.7.3.tgz.sha256`
- **Checksum for images' tgz file:** `occnp-images-1.7.3.tar.sha256`

2. Load the **occnp-images-1.7.3.tar** file into the Docker system

```
docker load --input /IMAGE_PATH/occnp-images-1.7.3.tar
```

3. Verify that the image is loaded correctly by entering this command:

```
docker images
```

ReferDocker Images for more information on docker images available in Cloud Native Core Policy (CNC Policy).

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

```
docker tag occnp/app_info:1.7.3 CUSTOMER_REPO/app_info:1.7.3
docker push CUSTOMER_REPO/app_info:1.7.3

docker tag occnp/oc-policy-ds:1.7.3 CUSTOMER_REPO/oc-policy-ds:1.7.3
docker push CUSTOMER_REPO/oc-policy-ds:1.7.3

docker tag occnp/ocingress_gateway:1.7.4 CUSTOMER_REPO/
ocingress_gateway:1.7.4
docker push CUSTOMER_REPO/ocingress_gateway:1.7.4

docker tag occnp/oc-pcf-sm:1.7.3 CUSTOMER_REPO/oc-pcf-sm:1.7.3
docker push CUSTOMER_REPO/oc-pcf-sm:1.7.3

docker tag occnp/oc-pcf-am:1.7.3 CUSTOMER_REPO/oc-pcf-am:1.7.3
docker push CUSTOMER_REPO/oc-pcf-am:1.7.3

docker tag occnp/oc-pcf-ue:1.7.3 CUSTOMER_REPO/oc-pcf-ue:1.7.3
docker push CUSTOMER_REPO/oc-pcf-ue:1.7.3

docker tag occnp/oc-audit:1.7.3 CUSTOMER_REPO/oc-audit:1.7.3
docker push CUSTOMER_REPO/oc-audit:1.7.3

docker tag occnp/oc-ldap-gateway:1.7.3 CUSTOMER_REPO/oc-ldap-
gateway:1.7.3
```

```
docker push CUSTOMER_REPO/oc-ldap-gateway:1.7.3

docker tag occnp/oc-query:1.7.3 CUSTOMER_REPO/oc-query:1.7.3
docker push CUSTOMER_REPO/oc-query:1.7.3

docker tag occnp/oc-pre:1.7.3 CUSTOMER_REPO/oc-pre:1.7.3
docker push CUSTOMER_REPO/oc-pre:1.7.3

docker tag occnp/oc-perf-info:1.7.3 CUSTOMER_REPO/oc-perf-info:1.7.3
docker push CUSTOMER_REPO/oc-perf-info:1.7.3

docker tag occnp/oc-diam-gateway:1.7.3 CUSTOMER_REPO/oc-diam-
gateway:1.7.3
docker push CUSTOMER_REPO/oc-diam-gateway:1.7.3

docker tag occnp/oc-diam-connector:1.7.3 CUSTOMER_REPO/oc-diam-
connector:1.7.3
docker push CUSTOMER_REPO/oc-diam-connector:1.7.3

docker tag occnp/oc-pcf-user:1.7.3 CUSTOMER_REPO/oc-pcf-user:1.7.3
docker push CUSTOMER_REPO/oc-pcf-user:1.7.3

docker tag occnp/oc-config-mgmt:1.7.3 CUSTOMER_REPO/oc-config-
mgmt:1.7.3
docker push CUSTOMER_REPO/oc-config-mgmt:1.7.3

docker tag occnp/oc-config-server:1.7.3 CUSTOMER_REPO/oc-config-
server:1.7.3
docker push CUSTOMER_REPO/oc-config-server:1.7.3

docker tag occnp/ocegress_gateway:1.7.7 CUSTOMER_REPO/
ocegress_gateway:1.7.7
docker push CUSTOMER_REPO/ocegress_gateway:1.7.7

docker tag occnp/nrf-client:1.2.5 CUSTOMER_REPO/nrf-client:1.2.5
docker push CUSTOMER_REPO/nrf-client:1.2.5

docker tag occnp/oc-readiness-detector:1.7.3 CUSTOMER_REPO/oc-
readiness-detector:1.7.3
docker push CUSTOMER_REPO/oc-readiness-detector:1.7.3

docker tag occnp/configurationinit:1.2.0 CUSTOMER_REPO/
configurationinit:1.2.0
docker push CUSTOMER_REPO/configurationinit:1.2.0

docker tag occnp/configurationupdate:1.2.0 CUSTOMER_REPO/
configurationupdate:1.2.0
docker push CUSTOMER_REPO/configurationupdate:1.2.0

docker tag occnp/oc-soap-connector:1.7.3 CUSTOMER_REPO/occnp/oc-
soap-connector:1.7.3
docker push CUSTOMER_REPO/occnp/oc-soap-connector:1.7.3

docker tag occnp/oc-pcrf-core:1.7.3 CUSTOMER_REPO/occnp/oc-pcrf-
core:1.7.3
```

```
docker push CUSTOMER_REPO/occnp/oc-pcrf-core:1.7.3

docker tag occnp/oc-binding:1.7.3 CUSTOMER_REPO/occnp/oc-
binding:1.7.3
docker push CUSTOMER_REPO/occnp/oc-binding:1.7.3
```

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

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

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

Upgrading CNC Policy (1.7.x to 1.7.3)

To upgrade:

1. Modify the required custom-values.yaml file with the required input parameters. To customize the file, see Customizing Cloud Native Core Policy.

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

> **Note:**
>
> **perf-info** has to be provided proper URL or else it will keep on restarting. [Below is an example of URL for bastion server]:
>
> ```
> perf-info:
>
> configmapPerformance:
>
> prometheus: http://occne-prometheus-server.occne-infra.svc
> ```

2. > **Caution:**
   >
   > Do not exit from `helm upgrade` command manually. After running the `helm upgrade` command, it takes some time to install all the services. In the meantime, you must not press "ctrl+c" to come out from `helm upgrade` command. It leads to some anomalous behavior.

   a. Upgrade CNC Policy by using Helm2:

   ```
   helm upgrade <release-namespace> <helm-chart> -f <custom-file>
   ```

   b. Upgrade CNC Policy by using Helm3:

   ```
   helm upgrade <release-name> <helm-chart> -f <custom-file> -n
   <release-namespace>
   ```

   where:
   *helm_chart*is the location of the helm chart extracted from occnp-pkg-1.7.3.tgz file

   release_name is the release name used by helm command.

   release_namespace is the deployment namespace used by helm command.

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

3. Execute the following command to get status of jobs and pods:

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

   For example:

   ```
   kubectl get pod -n occnp
   ```

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

E

Downgrading Cloud Native Core Policy

This chapter describes the Cloud Native Core Policy (CNC Policy) roll back procedure from CNC Policy 1.7.3 to previous version.

> **Note:**
>
> You can roll back maximum to last three releases. To downgrade CNC Policy to an older version, you must restore the configurations from backup.

To roll back from CNC Policy 1.7.3 to previous version:

1. Check which revision you need to roll back by executing the below command:

   ```
   helm history <release_namespace>
   ```

2. Execute the roll back command to roll back to that revision:
   a. Below is a command to roll back using Helm2:

      ```
      helm rollback <release_namespace> <revision number>
      ```

   b. Below is a command to roll back using Helm3:

      ```
      helm rollback <release_name> <revision number> -n
      <release_namespace>
      ```