Oracle® Communications

Network Analytics Data Director Installation, Upgrade, and Fault Recovery Guide

Release 25.2.200
G49042-01
December 2025

Oracle Communications Network Analytics Data Director Installation, Upgrade, and Fault Recovery Guide, Release 25.2.200

G49042-01

Contents

3 Customizing OCNADD

4 Upgrading OCNADD

Preface

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

My Oracle Support (MOS)

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

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

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

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

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

Acronyms

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

Table Acronyms and Terminology

Acronym	Definition
CA	Certificate Authority
CNC Console	Cloud Native Configuration Console
CNLB	Cloud Native Load Balancer
CLI	Command Line Interface
CN	Common Name
CSP	Communication Service Provider
OKE	Container Engine for Kubernetes
KPI	Key Performance Indicator
MPS	Messages Per Second
MOS	My Oracle Support
NDB	Network Data Broker
NF	Network Function
OCI	Oracle Cloud Infrastructure
OCCM	Oracle Communication Certificate Manager
BSF	Oracle Communications Cloud Native Core, Binding Support Function
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
NRF	Oracle Communications Cloud Native Core, Network Repository Function (NRF)
PCF	Oracle Communications Cloud Native Core, Policy Control Function
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy
NWDAF	Oracle Communications Networks Data Analytics Function
OHC	Oracle Help Center
OSDC	Oracle Service Delivery Cloud
SVC	Services
SAN	Subject Alternate Name
TLS	Transport Layer Security
URI	Uniform Resource Identifier

What's New in This Guide

This section lists the documentation updates for Release *25.2.2xx* in *Oracle Communications Network Analytics Data Director Installation, Upgrade, and Fault Recovery Guide*.

Release 25.2.200 - G49042-01, December 2025

Major Architecture Restructuring: Data Director Group Classification

This release introduces a fundamental transformation in the OCNADD architecture. The previously singular **Worker Group** has been decoupled and re-architected into two specialized functional groups: the **Relay Agent Group** and the **Mediation Group**.

This architectural shift impacts the entire system workflow and is reflected throughout this document:

- **Group-Specific installation, configuration and uninstallation procedures:**
  - Configuration procedures are no longer generic to a "Worker Group." All setup procedures are now distinct for **Relay Agent** nodes and **Mediation** nodes or namespaces.
- **Customization:**
  - Several unsupported parameters have been removed.
  - Parameters have been classified for each group.
- **Upgrades and Rollback:**
  - Upgrade is replaced with a migration procedure.
  - In-service rollback is not supported in this release.
- **Fault Recovery:**
  - Several procedures have been updated for the new architecture.

1 Introduction

This chapter provides information about installing Oracle Communications Network Analytics Data Director (OCNADD) and its microservices on the supported platforms.

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

1.1 Overview

With the emergence of 5G networks, Communication Service Providers (CSPs) now have access to vast amounts of data. Oracle Communications Network Analytics Data Director (OCNADD) serves as a specialized Network Data Broker (NDB) within the 5G Network Architecture. It collects network traffic data from various sources such as 5G network functions (NFs), non-5G NFs, and third-party producers. OCNADD then performs a range of rule-based operations to help CSPs maximize the value of this data. These operations include data aggregation, data filtering, data replication, data governance, and secure data transmission for subscribed third-party consumers.

By efficiently collecting and utilizing data, OCNADD provides CSPs with the following advantages:

- Enhanced service quality
- Ease of scalability
- Simplified monitoring and troubleshooting
- Identification of new revenue and network monetization opportunities
- Reduced network downtime

As an NDB, OCNADD sits between the 5G infrastructure and third-party tools or consumer applications. Its primary function is to ensure data security, low latency, and redundancy while efficiently collecting and processing data. By correlating and transforming acquired data based on configurable data feed settings, OCNADD enables CSPs to generate comprehensive dashboards and Key Performance Indicators (KPIs). These insights provide a deep understanding of all functions within the 5G Network Architecture, allowing CSPs to improve service quality, reduce downtime, support network scalability, and minimize losses.

In the event of network failures, OCNADD data is used for monitoring and troubleshooting. Additionally, OCNADD offers a user-friendly GUI that supports the creation, editing, and deletion of data feeds. For more information about OCNADD architecture and features, see the *Oracle Communications Network Analytics Data Director User Guide*.

Deployment Overview

OCNADD deployment models help customers optimize or reduce the Data Director footprint based on the third-party consumer's capability to consume data directly from the Kafka cluster. Users can deploy different Data Director models by configuring the applicable custom values.

The OCNADD deployment consists of the following groups:

OCNADD Management Group:

The OCNADD Management Group serves as the central hub for configuration management, health monitoring, and alarm handling across all OCNADD services. It is also responsible for managing export and tracing for critical transactions within Data Director, ensuring comprehensive visibility and control.

OCNADD Worker Group:

The OCNADD Worker Group is a logical entity divided into two sub-groups:

- **OCNADD Relay Agent Group:** Responsible for receiving data from source Network Functions (NFs) and performing essential operations, including data aggregation, non-transaction-based filtering, message sequencing, and metadata enrichment.
- **OCNADD Mediation Group:** Receives processed data from the associated Relay Agent and applies further processing, including transaction-based filtering, correlation, storing XDRs in supported formats to centralized data stores, and forwarding data to third-party probes.

Both the Relay Agent Group and the Mediation Group can be linked to a worker group, which is configurable through Helm charts. Within this worker group, the Mediation service can be associated with a specific Relay Agent. To establish this association, update the Mediation values file with the namespace and cluster details of the Relay Agent using Helm charts.

For more details, refer to the OCNADD Deployment Models section to understand the available deployment models and determine the required resources before proceeding with installation.

1.2 References

For more information on OCNADD, refer to the following documents:

- *Oracle Communications Network Analytics Suite Release Notes*
- *Oracle Communications Network Analytics Suite Licensing Information User Manual*
- *Oracle Communications Network Analytics Automated Testing Suite Guide*
- *Oracle Communications Network Analytics Suite Security Guide*
- *Oracle Communications Network Analytics Data Director User Guide*
- *Oracle Communications Network Analytics Data Director Outbound Interface Specification Guide*
- *Oracle Communications Network Analytics Data Director Benchmarking Guide*
- *Oracle Communications Network Analytics Data Director Diameter User Guide*
- *Oracle Communications Network Analytics Data Director vCollector Installation Guide*
- *Oracle Communications Network Analytics Data Director Troubleshooting Guide*

- *Oracle Communications Cloud Native Core, cnDBTier Installation,Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, OCI Deployment Guide*
- *Oracle Communication Certificate Manager Installation, Upgrade and Fault Recovery Guide*
- *Oracle Communication Certificate Manager User Guide*

1.3 Oracle Error Correction Policy

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

Table 1-1 Oracle Error Correction Policy

Release Number	General Availability (GA) Date	Error Correction Grace Period End Date
25.2.200	December 2025	December 2026
25.2.100	September 2025	September 2026
25.1.200	July 2025	July 2026
25.1.100	February 2025	February 2026

Note

- For the latest patch releases, see their corresponding *Oracle Communications Cloud Native Core Release Notes*.
- For a release, Sev1 and Critical Patch Unit (CPU) patches are supported for 12 months. For more information, see the Oracle Communications Cloud Native Core and Network Analytics Error Correction Policy.

1.4 Oracle Open Source Support Policies

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

2 Installing OCNADD

This chapter provides information about installing Oracle Communications Network Analytics Data Director (OCNADD) on the supported platforms.

The OCNADD installation is supported over the following platforms:

- Oracle Communications Cloud Native Core, Cloud Native Environment (CNE)
- VMware Tanzu Application Platform (TANZU)
- Oracle Cloud Infrastructure (OCI)

> **Note**
>
> This document describes the OCNADD installation on CNE. However, the procedure for installation on OCI and TANZU is similar to the installation on CNE. Any steps unique to OCI or TANZU platform are mentioned explicitly in the document.

2.1 Prerequisites

Before installing and configuring OCNADD, make sure that the following requirements are met:

2.1.1 Software Requirements

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

Table 2-1 Mandatory Software

Software	Version
Kubernetes	1.33.x, 1.32.x
Helm	3.15.2
Docker/Podman	4.6.1
OKE (on OCI)	1.27.x

> **Note**
>
> - OCNADD 25.2.200 supports CNE 25.2.1xx and 25.1.2xx.

To check the Oracle Communications Cloud Native Core, Cloud Native Environment (CNE) version, run the following command:

```
echo $OCCNE_VERSION
```

To check the current Helm and Kubernetes versions installed in CNE, run the following commands:

```
kubectl version
```

```
helm version
```

> **Note**
>
> - Starting with CNE 1.8.0, Podman is the preferred container platform instead of docker. For more information on installing and configuring Podman, see the *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

If you are installing OCNADD on TANZU, the following software must be installed:

Table 2-2 Mandatory Software

Software	Version
Tanzu	1.4.1

To check the current TANZU version, run the following command:

```
tanzu version
```

> **Note**
>
> Tanzu was supported in release 22.4.0. Release 25.2.200 has not been tested on Tanzu.

Depending on the requirement, you may have to install additional software while deploying OCNADD. The list of additional software items, along with the supported versions and usage, is given in the following table:

Table 2-3 Additional Software

Software	Version	Required For
Prometheus-Operator	2.52.0	Metrics
Metallb	0.14.4	LoadBalancer
cnDBTier	25.2.1xx and 25.1.2xx	MySQL Database
Druid	33.0.0	It is required for extended storage integration with the Druid database.

> **Note**
>
> - Some of the software are available by default if OCNADD is getting deployed in Oracle Communications Cloud Native Core, Cloud Native Environment (CNE).
> - Install the additional software if any of them are not available by default with CNE.
> - If you are deploying OCNADD in any other environment, for instance, TANZU, then all the above mentioned software must be installed before installing OCNADD.
> - On OCI, the Prometheus-Operator is not required. The metrics and alerts will be managed using OCI monitoring and Alarm services.

To check the installed software items, run the following command:

```
helm ls -A
```

2.1.2 Environment Setup Requirements

This section provides information on environment setup requirements for installing Oracle Communications Network Analytics Data Director (OCNADD).

Network Requirements

The Data Director services, such as Kafka and Redundancy Agent, require external access. These services are created as load balancer services, and the service FQDNs should be used for communication with them. Additionally, the service FQDNs must be configured in the DNS server.

CNLB Network and NADs for Data Director

Egress NADs

1. Customer must know or create Egress NADs for its third-party feed endpoint requirements before CNLB CNE cluster installation. The Egress NADs are required to be defined in the `cnlb.ini` file of OCCNE for the CNLB support.
2. The Egress NADs are required to be created for the following traffic segregation scenarios:
   a. Separate Egress NAD per third-party destination endpoint per third-party feed: Each destination endpoint of the consumer adapter will have its separate egress network via a separate Egress NAD managed by CNLB.
   b. Separate Egress NAD per third-party feed: Each consumer adapter feed will have its separate egress network via a separate Egress NAD managed by CNLB.
   c. Separate Egress NAD per OCNADD: All the consumer adapter feeds will have only one separate network via a separate Egress NAD managed by CNLB.

Ingress NADs

1. Customer must know or create the required CNLB IPs (external IPs) and ingress OCNADDs for the *Data Director Ingress Adapter* service.
2. Based on the ingress traffic segregation requirement for non-Oracle NFs, the required CNLB IPs (external IPs) and ingress OCNADDs need to be configured for the Ingress Adapter in advance. The ingress OCNADDs are required to be defined in the `cnlb.ini` file of OCCNE for CNLB support.

3. Each Ingress Adapter service instance must have an external IP and a corresponding ingress OCNADD created and managed by the CNLB.
4. Customer must know or create the ingress OCNADD for the redundancy agent external access and IP.
5. Customer must know or create the required CNLB IPs (external IPs) and ingress OCNADDs for the *Data Director Kafka* service. The number of ingress OCNADDs and external IPs must be the same as the number of Kafka brokers in the cluster. This must be done for every additional worker group that is present or needs to be created in the future.
6. The required CNLB external IP and corresponding ingress OCNADD must be configured in the `cnlb.ini` file of OCCNE for CNLB support.

Ingress-Egress NADs

- Customer must know or create the required CNLB IPs (external IPs) and ingress-egress NADs for the Data Director Gateway service when external access is enabled for gateway services.
- Gateway service present in each NAD group must have an external IP and a corresponding ingress-egress NAD created and managed by the CNLB.
- The required CNLB external IP and corresponding ingress-egress NAD must be configured in the `cnlb.ini` file of OCCNE for CNLB support.

For more information on the CNLB and NADs, refer to the *Oracle Communications Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

Environment Setup on OCCNE

Network Access

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

1. **Local container image repository**: It contains the OCNADD container images. To check if the Kubernetes cluster hosts can access the local container image repository, pull any image with an image-tag using the following command:

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

   where,

   - `docker-repo` is the IP address or hostname of the container image repository.
   - `image-name` is the container image name.
   - `image-tag` is the tag assigned to the container image used for the OCNADD pod.
2. **Local Helm repository**: It contains the OCNADD Helm charts. To check if the Kubernetes cluster hosts can access the local Helm repository, run the following command:

   ```
   helm repo update
   ```

3. Service FQDN or IP Addresses of the required OCNADD services, for instance, Kafka Brokers, must be discoverable from outside of the cluster. This information should be publicly exposed so that Ingress messages to OCNADD can come from outside of Kubernetes.

Environment Setup on OCI

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

For OCI infrastructure, see *Oracle Communications Cloud Native Core OCI Adaptor, NF Deployment on OCI Guide*.

After completing the OCI infrastructure setup requirements, proceed to the next section.

Client Machine Requirements

> **Note**
>
> Run all the `kubectl` and `helm` commands in this guide on a system depending on the infrastructure and deployment. This system could be a client machine, such as a virtual machine, server, local desktop, etc.

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

The client machine must meet the following requirements:

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

Server or Space Requirements

For information on the server or space requirements for installing OCNADD, see the following documents:

- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Network Analytics Data Director Benchmarking Guide*
- *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*

cnDBTier Requirement

> **Note**
>
> Obtain the values of the `cnDBTier` parameters listed in the section "cnDBTier Customization Parameters" from the delivered `ocnadd_dbtier_custom_values.yaml` file and use these values in the new `ocnadd_dbtier_custom_values.yaml` file if the parameter values in the new file differ from those in the delivered file.
>
> If you already have an older version of `cnDBTier`, upgrade `cnDBTier` with resources recommended for OCNADD by customizing the `ocnadd_dbtier_custom_values.yaml` file in the `custom_templates` folder of the OCNADD package with the required deployment parameters. Use the same PVC size as in the previous release. For more information, see the section "cnDBTier Customization Parameters."

OCNADD supports cnDBTier 25.2.1xx and 25.1.2xx in a CNE environment. cnDBTier must be up and running before installing the Data Director. To install cnDBTier 25.2.2xx with resources recommended for OCNADD, customize the `ocnadd_dbtier_custom_values.yaml` file in the custom_templates folder in the OCNADD package with the required deployment parameters.

> **Note**
>
> The `ocnadd_dbtier_custom_values.yaml` file in the DD custom_templates.zip should normally correspond to the same version as the Data Director; however, it may be possible that the cnDBTier custom values belong to a different version than the Data Director. In this case, the `global.version` parameter from the `ocnadd_dbtier_custom_values.yaml` should be checked, and the corresponding GA package of cnDBTier should be used for the installation or upgrade of cnDBTier before installing/upgrading the Data Director package.

cnDBTier parameters for the Data Director may vary. For more information, see section cnDBTier Customization Parameters**.**

For more information about the cnDBTier installation procedure, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

> **Note**
>
> For OCI Environment, use the `StorageClass` as `oci-bv` in cnDBTier charts. To find the storage class name, run the below command:
>
> ```
> kubectl get sc -n <namespace>
> ```

2.1.3 Capacity Planning

2.1.3.1 OCNADD Deployment Models

OCNADD supports the following deployment models:

- **Model 1:** OCNADD Management Group Services, Relay Agent Group Services and Mediation Group Services in same cluster
- **Model 2:** OCNADD Management Group Services, Relay Agent Group Services and Mediation Group Services in different cluster
- **Model 3:** OCNADD Management Group Services, Relay Agent Group Services and Mediation Group with Kafka only

> **Note**
>
> The Data Director supports egress adapters for outbound connections. The egress adapters add value to the message feed by filtering and synthesizing the packets before sending the messages out on the egress connection type 'HTTP/2' or 'Synthetic Feed'. If the customer selects a deployment model that does not include the Egress adapter, additional features such as synthetic packet generation will not be available, although the filtering and correlation features will be available using Kafka feeds only.

2.1.3.1.1 Model 1: OCNADD Management Group Services, Relay Agent Group Services and Mediation Group Services in Same Cluster

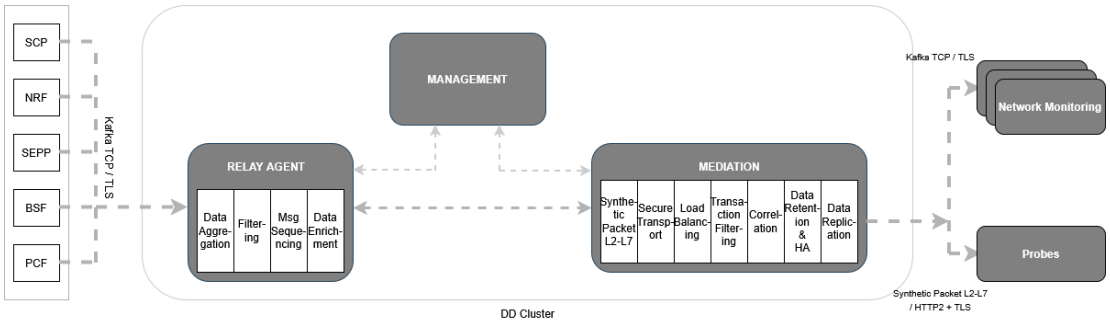
This OCNADD deployment model includes all the services deployed in the same cluster. For each OCNADD group, separate namespaces must be created during the deployment. In this deployment option, all the features are available. This is the default model, and the required services are enabled by default in each OCNADD group custom values file (`ocnadd-<ocnadd-group>-custom-values-25.2.200.yaml`).

In this deployment model, the default Kafka storage options are as follows:

- Relay agent Kafka is enabled with the Volatile (RAM Drive) storage option.
- Mediation Kafka is deployed with the Persistence (Disk) storage option.

Use the UI to configure message feeds on OCNADD. The Oracle Producer NFs (SCP, NRF, SEPP, BSF, and PCF) copy the messages to their respective source topics.

Figure 2-1 Model 1



For this model, the user only needs to enable the required aggregation services in the `ocnadd-relayagent-custom-values-25.2.200.yaml`. The default parameters are as below.

```
---
global:
  ocnaddmanagement:
    ocnaddalarm:
      enabled: true
    ocnaddconfiguration:
      enabled: true
    ocnaddhealthmonitoring:
      enabled: true
    ocnaddbackuprestore:
      enabled: true
    ocnadduirouter:
      enabled: true
    ocnaddgui:
      enabled: true
    ocnaddexport:
      enabled: false
    ocnaddmanagementgateway:
      enabled: true
---
global:
  ocnaddrelayagent:
    ocnaddscpaggregation:
      enabled: true
    ocnaddseppaggregation:
      enabled: false       # Enable to 'true' if data streaming from SEPP is
required
    ocnaddnrfaggregation:
      enabled: false       # Enable to 'true' if data streaming from NRF is
required
    ocnaddbsfaggregation:
      enabled: false       # Enable to 'true' if data streaming from BSF is
required
    ocnaddpcfaggregation:
      enabled: false       # Enable to 'true' if data streaming from PCF is
required
    ocnaddkafka:
      enabled: true
    ocnaddrelayagentgateway:
      enabled: true
---
global:
  ocnaddmediation:
    ocnaddkafka:
      enabled: true
    ocnaddadmin:
      enabled: true
    ocnaddfilter:
      enabled: false
    ocnaddmediationgateway:
      enabled: true
```

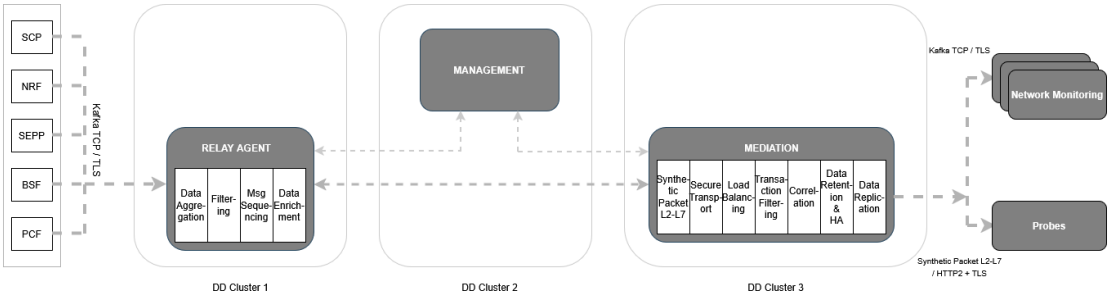
2.1.3.1.2 Model 2: OCNADD Management Group Services, Relay Agent Group Services and Mediation Group Services in Different Clusters

In this OCNADD deployment model, all the OCNADD groups, for example the Management group, Relay agent group and Mediation group, are deployed in different clusters. In this deployment option, all the features are available.

For this deployment model, the default Kafka storage options are as follows:

- Relay agent Kafka is enabled with the Volatile (RAM Drive) storage option.
- Mediation Kafka is deployed with the Persistence (Disk) storage option.

Figure 2-2 Model 2



To utilise this deployment option, ensure that mTLS is enabled across OCNADD for all groups. Additionally, configure all gateways and the mediation Kafka cluster with external access enabled.

For detailed instructions on enabling *External Communication Between Gateways* and *Enable External Access For Kafka Cluster*, refer to the *Oracle Communications Network Analytics Data Director User Guide*.

In this deployment mode, various combinations are supported. Some possible deployment combinations include:

- Management Group Services and Relay Agent Group Services are deployed in the same cluster, and Mediation Group Services are deployed in a different cluster (external access for Mediation Kafka must be enabled).
- Management Group Services and Mediation Group Services are deployed in the same cluster, and Relay Agent Group Services are deployed in a different cluster (external access for Mediation Kafka must be enabled).
- Relay Agent Group Services and Mediation Group Services are deployed in the same cluster, and Management Group Services are deployed in a different cluster.

Recommendation

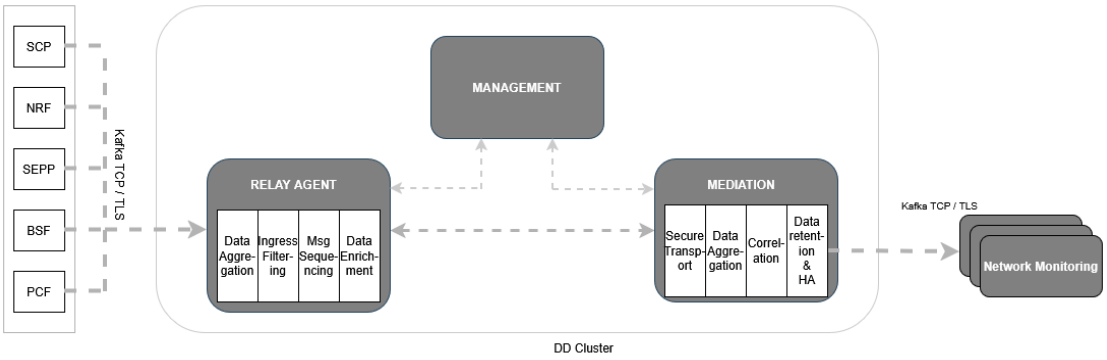
This deployment option is recommended only when the target cluster lacks the necessary hardware resources or has suboptimal disk throughput. Note that this configuration may result in higher end-to-end latency.

2.1.3.1.3 Model 3: OCNADD Management Group Services, Relay Agent Group Services and Mediation Group with Kafka Only

Use this model when the customer does not wish to receive the message feed using HTTP/2 or TCP connection mode. The third-party monitoring application available to the customer can consume data directly from the Kafka cluster. The Egress adapter is not required in this deployment model; however, the OCNADD deployment requires common services such as UI, Configuration, Health Monitoring, Alarm, and Admin. Features like correlation-id-based load balancing, synthetic feed, and HTTP/2 feeds are unavailable in this deployment model, although the filtering and correlation features will be available using Kafka feeds only.

This model saves the Egress adapter resource; however, additional resources will be required for Filtering and Correlation services once these features are used in the configurations from the UI. The export feature is also possible; however, it has to be enabled in the charts by enabling the `ocnaddexport` service and further export configuration from the UI.

Figure 2-3 Model 3



This deployment model supports a direct Kafka feed. For more information, see the *External Kafka Feeds* section in the *Oracle Communications Network Analytics Data Director User Guide*.

In this deployment option, Mediation Group Services can co-exist with Relay Agent Group Services and Management Group Services, or they can exist independently in a separate cluster. Higher end-to-end latency may be reported if the Relay Agent Group and Mediation Group are deployed in separate clusters.

The default Kafka storage options for this model are as follows:

- Relay agent Kafka is enabled with the Volatile (RAM Drive) storage option.
- Mediation Kafka is deployed with the Persistence (Disk) storage option.

The default parameters are as below.

```
---
global:
    ocnaddmanagement:
        # services provided for management
        ocnaddalarm:
            enabled: true
        ocnaddconfiguration:
```

```
            enabled: true
        ocnaddhealthmonitoring:
            enabled: true
        ocnaddbackuprestore:
            enabled: true
        ocnadduirouter:
            enabled: true
        ocnaddgui:
            enabled: true
        ocnaddexport:
            enabled: false                ## --> Enable to 'true' if XDR
Export or Trace feature is required
        ocnaddmanagementgateway:
            enabled: true
---
global:
    ocnaddrelayagent:
        ocnaddscpaggregation:
            enabled: true
        ocnaddseppaggregation:
            enabled: false               ## --> Enable to 'true' if data
streaming from SEPP is required
        ocnaddnrfaggregation:
            enabled: false               ## --> Enable to 'true' if data
streaming from NRF is required
        ocnaddbsfaggregation:
            enabled: false               ## --> Enable to 'true' if data
streaming from BSF is required
        ocnaddpcfaggregation:
            enabled: false               ## --> Enable to 'true' if data
streaming from PCF is required
        ocnaddkafka:
            enabled: true
        ocnaddrelayagentgateway:
            enabled: true
---
global:
    ocnaddmediation:
        ocnaddkafka:
            enabled: true
        ocnaddadmin:
            enabled: true
        ocnaddfilter:
            enabled: false               ## --> Enable to 'true' if FILTERED
or CORRELATED_FILTERED kafka Feeds are required
        ocnaddmediationgateway:
            enabled: true
---
```

1. The aggregation service aggregates traffic from the source topics to the Kafka main topic. Choosing any specific combination of NFs for aggregation rules is not possible. The total traffic received is aggregated and available to the consumers.
2. The third-party consumer application must create external Kafka feeds to connect with the Kafka cluster, which allows it to consume messages directly from the designated topic.

2.1.3.2 Resource Comparison

The following table depicts the resource savings in the various deployment models:

Table 2-4 Resource Comparison

Deployment Model	Model 1	Model 2	Model 3
Common Services	Available	Available	Available
Aggregation Service	Available	Available	Available
Adapter Service	Available	Available	Not Available
Kafka	Available	Available	Available
Resource Saving (approx. %)	0	0	60
Supported Egress Interfaces	HTTP/2 TCP KAFKA	HTTP/2 TCP KAFKA	KAFKA

The customer can customize the OCNADD deployment based on the identified resources. Plan the resources based on the deployment model and services required for the specific model.

2.1.3.3 Key Points to Consider for All Deployment Models

1. The message feeds must be created from the UI, and aggregation rules determine the source NF combinations for aggregation.
2. Metrics related to the feed are available on the UI.
3. OCNADD alarms can be viewed on the UI.

2.1.3.4 Kafka Storage Mode Comparison

The following table outlines the benefits of different Kafka storage options. Choose the one that meets your requirements:

Table 2-5 Kafka Storage Mode Comparison

Parameters	Relay Agent Kafka Persistence Storage (Disk) – Mediation Kafka Persistence Storage (Disk)	Relay Agent Kafka Volatile Storage (RAM Drive) – Mediation Kafka Persistence Storage (Disk) (Default)	Relay Agent Kafka Volatile Storage (RAM Drive) – Mediation Kafka Volatile Storage (RAM Drive)
Throughput	Data processing speed is largely limited by disk read/write performance	Delivers high throughput; the relay-agent Kafka runs on a RAM drive for faster I/O, while disk is used only by the Mediation Kafka, reducing overall disk usage	Delivers very high throughput as broker processing and I/O latency are minimized

Table 2-5 (Cont.) Kafka Storage Mode Comparison

Parameters	Relay Agent Kafka Persistence Storage (Disk) – Mediation Kafka Persistence Storage (Disk)	Relay Agent Kafka Volatile Storage (RAM Drive) – Mediation Kafka Persistence Storage (Disk) (Default)	Relay Agent Kafka Volatile Storage (RAM Drive) – Mediation Kafka Volatile Storage (RAM Drive)
Latency	Higher latency than RAM drives due to slower disk read/write performance	Offers low latency, but overall latency is constrained by disk I/O since Mediation Kafka uses disk storage mode	Ultra-low traffic processing latency as RAM drive read/write can be in microseconds to low milliseconds as compared to Disk
Data Retention and Storage	Supports high data retention; achieving longer retention requires additional disk capacity however, it is less expensive than RAM drive. Recommended for use cases where data retention is a priority.	Supports high data retention since Mediation Kafka can store data longer, but it requires additional disk capacity. Preferable for workloads where throughput and retention are of coequal priorities	Can support higher data retention, but requires substantially more RAM, which can be expensive. Best suited for scenarios with low retention needs.

During installation, user can choose between RAM and CEPH storage modes for both the Relay Agent and Mediation Kafka clusters based on the requirements.

2.1.3.5 Source NF and OCNADD Relay Agent Kafka Access Modes

Choose the Relay Agent Kafka access mode for forwarding traffic between source Network Functions and OCNADD.

The following access modes are supported for the Relay Agent Kafka broker:

NF producers and OCNADD Relay Agent Kafka in the same cluster

In this mode, the Kafka cluster is not exposed externally. By default, the parameters to enable external access for Kafka are set to `false`, hence no changes are required.

- All three ports can be used, for example `9092` for `PLAIN_TEXT`, `9093` for `SSL`, and `9094` for `SASL_SSL`. However, the `9092` port is non-secure and therefore not recommended for use.
- It is recommended to configure individual broker IPs/FQDNs in the Kafka bootstrap server list.

```
kafka-broker-0.kafka-broker-headless.<namespace>.svc.<domain>:9093/9092
kafka-broker-1.kafka-broker-headless.<namespace>.svc.<domain>:9093/9092
kafka-broker-2.kafka-broker-headless.<namespace>.svc.<domain>:9093/9092
kafka-broker-3.kafka-broker-headless.<namespace>.svc.<domain>:9093/9092
```

NF producers and OCNADD Relay Agent Kafka in different clusters

In this mode, the user must enable external access to the Kafka cluster using a `LoadBalancer` service type. Certificates must also be created with the LoadBalancer IP addresses assigned to the broker. The user can create certificates manually, using the generate-certs script, or through OCCM. For more details, see the *Oracle Communications Network Analytics Suite User Guide*, specifically the *Enable External Access For Kafka Cluster* section.

2.1.3.6 Resource Requirements

This section describes the resource requirements to install and run Oracle Communications Network Analytics Data Director (OCNADD).

OCNADD deployment consists of a management group and worker group(s). Traffic processing services are managed within the worker group, while configuration and administration services are managed within the management group.

Resource planning for OCNADD should consider the following points:

- There will be only one management group consisting of the following services:
  - ocnaddconfiguration
  - ocnaddalarm
  - ocnaddhealthmonitoring
  - ocnaddui
  - ocnadduirouter
  - ocnaddredundancyagent
  - ocnaddexport
  - ocnaddmanagementgateway
- The Worker Group is administered by the Management Group. A worker group is considered a logical entity that includes the following two OCNADD sub-groups and their respective services:
  - **Relay Agent Group**
    - ocnaddkafka
    - kraft-controller
    - ocnaddnrfaggregation
    - ocnaddseppaggregation
    - ocnaddscpaggregation
    - ocnaddpcfaggregation
    - ocnaddbsfaggregation
    - ocnaddrelayagentgateway
  - **Mediation Group**
    - ocnaddkafka
    - kraft-controller
    - ocnaddcorrelation
    - ocnaddfilter
    - ocnaddadmin
    - ocnaddconsumeradapter
    - ocnaddstorageadapter
    - ocnaddmediationgateway

- The customer needs to plan for the resources corresponding to the management group and worker group services required.

OCNADD Resource Requirements

This following default profile can stream data from NFs up to 15K MPS and can be scaled to handle up to 100K MPS for HTTP2/Synthetic feed when "weighted_lb" feature is not enabled.

Table 2-6 OCNADD Resource Requirements (All DD features with Default profile)

OCNADD Services	vCPU Req	vCPU Limit	Memory Req (Gi)	Memory Limit (Gi)	Min Replica	Max Replica	Partitions	Topic Name
Management Services								
ocnaddconfiguration	1	1	1	1	1	1	-	-
ocnaddalarm	1	1	1	1	1	1	-	-
ocnaddhealthmonitoring	1	1	1	1	1	1	-	-
ocnaddgui	1	2	1	1	1	2	-	-
ocnadduirouter	1	2	1	1	1	2	-	-
ocnaddredundancyagent	2	2	3	3	1	4	-	-
ocnaddexport	2	4	4	64	1	2	-	-
ocnaddmanagementgateway	1	1	1	1	1	2	-	-
Relay Agent Services								
ocnaddkafka	6	6	64	64	4	4	-	-
kraftcontroller	1	1	2	2	3	3	-	-
ocnaddscpaggregation	2	2	2	2	1	3	18	SCP
ocnaddnrfaggregation	2	2	2	2	1	1	6	NRF
ocnaddseppaggregation	2	2	2	2	1	2	12	SEPP
ocnaddpcfaggregation	2	2	2	2	1	2	12	PCF
ocnaddbsfaggregation	2	2	2	2	1	1	6	BSF
ocnaddrelayagentgateway	1	1	1	1	1	2	-	-
Mediation Services								
ocnaddadminservice	1	1	1	1	1	1	-	-
<app-name>-adapter	3	3	4	4	2	14	126	MAIN
ocnaddkafka	6	6	64	64	4	4	-	-
kraftcontroller	1	1	2	2	3	3	-	-
ocnaddcorrelation	3	3	24	64	1	4	-	-
ocnaddfilter	2	2	3	3	1	4	-	-
ocnaddstorageadapter	3	3	24	64	1	4	-	-
ocnaddingressadapter	3	3	8	8	1	7	-	-
ocnaddmediationgateway	1	1	1	1	1	2	-	-

> **Note**
>
> For detailed information on the OCNADD profiles, see the "Profile Resource Requirements" section in the *Oracle Communications Network Analytics Data Director Benchmarking Guide*.

Ephemeral Storage Requirements

Table 2-7 Ephemeral Storage

Service Name	Ephemeral Storage (min) in Mi	Ephemeral Storage (max) in Mi
Management Services		
ocnaddconfiguration	100	1000
ocnaddalarm	100	500
ocnaddhealthmonitoring	100	500
ocnaddredundancyagent	100	500
ocnaddexport	100	2Gi
ocnaddmanagementgateway	100	500
ocnadduirouter	500	500
Relay Agent Services		
ocnaddscpaggregation	500	500
ocnaddnrfaggregation	500	500
ocnaddseppaggregation	500	500
ocnaddpcfaggregation	500	500
ocnaddbsfaggregation	500	500
ocnaddrelayagentgateway	100	500
Mediation Services		
ocnaddadminservice	100	200
<app-name>-adapter	1000	1000
ocnaddcorrelation	100	500
ocnaddfilter	100	500
ocnaddstorageadapter	400	800
ocnaddingressadapter	400	800
ocnaddmediationgateway	100	500

2.2 Installation Sequence

This section provides information on how to install Oracle Communications Network Analytics Data Director (OCNADD).

> **Note**
>
> - It is recommended to follow the steps in the given sequence for preparing and installing OCNADD.
> - Make sure you have the required software installed before proceeding with the installation.
> - This is the installation procedure for a standard OCNADD deployment. To install a more secure deployment (such as, adding users, changing password, enabling mTLS, and so on) see, *Oracle Communications Network Analytics Suite Security Guide*.

2.2.1 Pre-Installation Tasks

To install OCNADD, perform the preinstallation steps described in this section.

> **Note**
>
> The `kubectl` commands may vary based on the platform used for deploying OCNADD. Users are recommended to replace `kubectl` with the environment-specific command-line tool used to configure Kubernetes resources through the kube-api server. The instructions provided in this document are as per the Oracle Communications Cloud Native Core, Cloud Native Environment (CNE) version of the kube-api server.

2.2.1.1 Downloading OCNADD Package

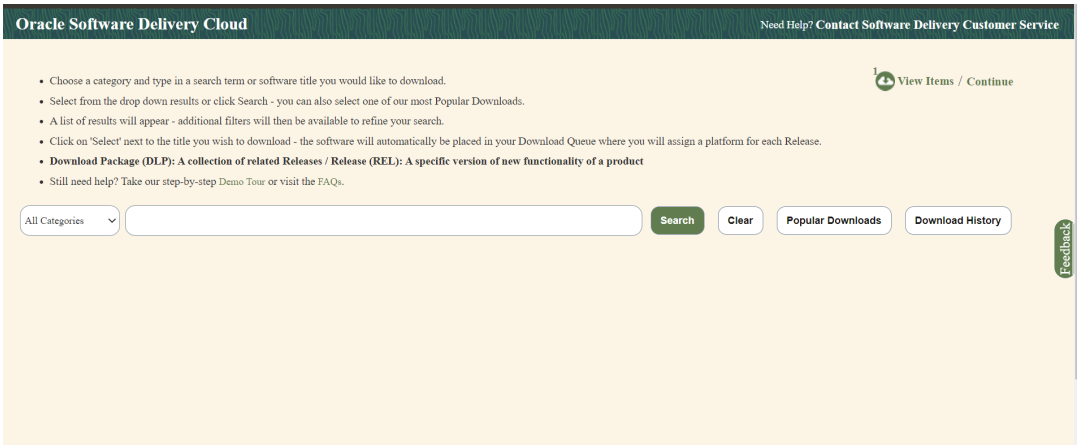
To download the Oracle Communications Network Analytics Data Director (OCNADD) package from MOS, perform the following steps:

1. Log in to My Oracle Support with your credentials.
2. Select the **Patches and Updates** tab to locate the patch.
3. In the **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter "Oracle Communications Network Analytics Data Director" in the **Product** field, select "Oracle Communications Network Analytics Data Director 25.2.200.0.0 from **Release** drop-down list.
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 download the OCNADD package file.
9. Extract the zip file to download the network function patch to the system where the network function must be installed.

To download the Oracle Communications Network Analytics Data Director package from the edelivery portal, perform the following steps:

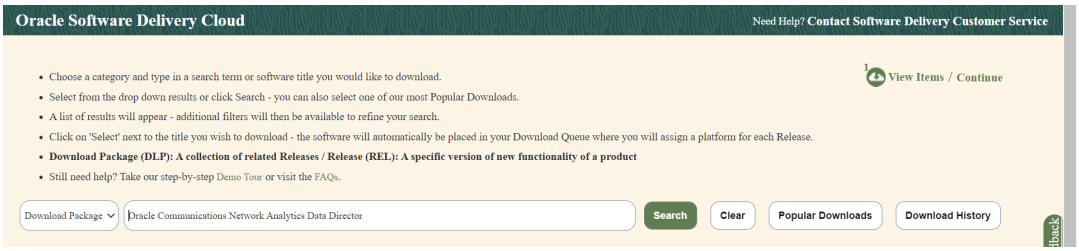
1. Login to the edelivery portal with your credentials. The following screen appears:

Figure 2-4 edelivery portal



2. Select the **Download Package** option, from **All Categories** drop down list.
3. Enter **Oracle Communications Network Analytics Data Director** in the search bar.

Figure 2-5 Search



4. List of release packages available for download are displayed on the screen. Select the release package you want to download, the package automatically gets downloaded.

2.2.1.2 Pushing the Images to Customer and OCI Registry

> **Note**
>
> The `kubectl` commands may vary based on the platform used for deploying OCNADD. Users are recommended to replace `kubectl` with the environment-specific command-line tool used to configure Kubernetes resources through the kube-api server. The instructions provided in this document are as per the Oracle Communications Cloud Native Core, Cloud Native Environment (CNE) version of the kube-api server.

Container Images

Oracle Communications Network Analytics Data Director (OCNADD) deployment package includes ready-to-use container images and helm charts to help orchestrate containers in Kubernetes. The communication between Pods of services of OCNADD are preconfigured in the helm charts.

The following table lists the container images of OCNADD. The table depicts the default OCNADD microservices and their respective images. However, a few more necessary images

are delivered as part of the OCNADD package; you must push these images along with the default images. The image names are suffixed with the OCNADD release name.

Table 2-8 Container Images for OCNADD

Service Name	Container Image Name	Image Tag
Management Services		
OCNADD-Configuration	ocnaddconfiguration	25.2.200
OCNADD-Alarm	ocnaddalarm	25.2.200
OCNADD-HealthMonitoring	ocnaddhealthmonitoring	25.2.200
OCNADD-UIRouter	ocnadduirouter	25.2.200
OCNADD-GUI	ocnaddgui	25.2.200
OCNADD-Redundancyagent	ocnaddredundancyagent	25.2.200
OCNADD-Export	ocnaddexport	25.2.200
OCNADD-ManagementGateway	ocnaddmanagementgateway	25.2.200
OCNADD-Backup-Restore	ocnaddbackuprestore	25.2.200
Relay Agent Services		
OCNADD-Kafka	kafka-broker-x	4.1.0:25.2.200
OCNADD-Aggregation	ocnaddnrfaggregation ocnaddscpaggregation ocnaddseppaggregation ocnaddnonoracleaggregation ocnaddpcfaggregation ocnaddbsfaggregation	25.2.200
OCNADD-RelayAgentGateway	ocnaddrelayagentgateway	25.2.200
Mediation Services		
OCNADD-Kafka	kafka-broker-x	4.1.0:25.2.200
OCNADD-Admin	ocnaddadminservice	25.2.200
OCNADD-ConsumerAdapter	ocnaddconsumeradapter	25.2.200
OCNADD-Filter	ocnaddfilter	25.2.200
OCNADD-Correlation	ocnaddcorrelation	25.2.200
OCNADD-StorageAdapter	ocnaddstorageadapter	25.2.200
OCNADD-IngressAdapter	ocnaddingressadapter	25.2.200
OCNADD-MediationGateway	ocnaddmediationgateway	25.2.200

Note

- The service image names are prefixed with the OCNADD release name.
- The above table depicts the default OCNADD microservices and their respective images. However, a few more necessary images are delivered as a part of the OCNADD package, make sure to push all the images delivered with the package.

Pushing OCNADD Images to Customer Registry

To push the images to the registry:

1. Untar the OCNADD package zip file to retrieve the OCNADD docker image tar file:

   ```
   tar -xvzf ocnadd_pkg_25_2_200.tar.gz

   cd ocnadd_pkg_25_2_200

   tar -xvzf ocnadd-25.2.200.tar.gz
   ```

   The directory consists of the following:

   - **OCNADD Docker Images File:**

     `ocnadd-images-25.2.200.tar`

   - **Helm File:**

     `ocnadd-25.2.200.tgz`

   - **Readme txt File:**

     `Readme.txt`

   - **Custom Templates:**

     `custom_templates.zip`

   - **ssl_certs folder:**

     `ssl_certs`

2. Run one of the following commands to first change the directory and then load the `ocnadd-images-25.2.200.tar` file:

   ```
   cd ocnadd-package-25.2.200
   ```

3. Run one of the following command to load the OCNADD images. Use the appropriate group name (management, relayagent, or mediation) in place of "<ocnadd-group>" for the images user is intended to load.

   ```
   docker load --input /IMAGE_PATH/ocnadd-images-25.2.200.tar


   podman load --input /IMAGE_PATH/ocnadd-images-25.2.200.tar
   ```

   Example: Considering podman for this example to load images

   ```
   podman load --input /IMAGE_PATH/ocnadd-management-images-25.2.200.tar
   podman load --input /IMAGE_PATH/ocnadd-relayagent-images-25.2.200.tar
   podman load --input /IMAGE_PATH/ocnadd-mediation-images-25.2.200.tar
   ```

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

   ```
   docker images
   ```

   ```
   podman images
   ```

   Verify the list of images shown in the output with the list of images shown in the table Table 2-8. If the list does not match, reload the image tar file.

5. Run one of the following commands to tag each imported image to the registry:

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

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

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

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

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

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

7. Run the following command to push the helm charts to the helm repository:

   ```
   helm push <image_name>.tgz <helm_repo>
   ```

8. Run the following command to extract the helm charts:

   ```
   tar -xvzf ocnadd-25.2.200.tgz
   ```

9. Run the following command to unzip the custom_templates.zip file.

   ```
   unzip custom_templates.zip
   ```

Pushing OCNADD Images to OCI Registry

To push the images to the registry:

1. Untar the OCNADD package zip file to retrieve the OCNADD docker image tar file:

   ```
   tar -xvzf ocnadd_pkg_25_2_200.tar.gz
   ```

   ```
   cd ocnadd_pkg_25_2_200
   ```

```
tar -xvzf ocnadd-25.2.200.tar.gz
```

The directory consists of the following:

- **OCNADD Docker Images File:**

  ```
  ocnadd-images-25.2.200.tar
  ```

- **Helm File:**

  ```
  ocnadd-25.2.200.tgz
  ```

- **Readme txt File:**

  ```
  Readme.txt
  ```

- **Custom Templates:**

  ```
  custom_templates.zip
  ```

- **ssl_certs folder:**

  ```
  ssl_certs
  ```

2. Run one of the following commands to first change the directory and then load the `ocnadd-images-25.2.200.tar` file:

   ```
   cd ocnadd-package-25.2.200
   ```

3. Run one of the following command to load the OCNADD images. Use the appropriate group name (management, relayagent, or mediation) in place of "<ocnadd-group>" for the images user is intended to load.

   ```
   docker load --input /IMAGE_PATH/ocnadd-images-25.2.200.tar
   ```

   ```
   podman load --input /IMAGE_PATH/ocnadd-images-25.2.200.tar
   ```

   Example: Considering podman for this example to load images

   ```
   podman load --input /IMAGE_PATH/ocnadd-management-images-25.2.200.tar
   podman load --input /IMAGE_PATH/ocnadd-relayagent-images-25.2.200.tar
   podman load --input /IMAGE_PATH/ocnadd-mediation-images-25.2.200.tar
   ```

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

   ```
   docker images
   ```

   ```
   podman images
   ```

Verify the list of images shown in the output with the list of images shown in the table Table 2-8. If the list does not match, reload the image tar file.

5. Run the following commands to log in to the OCI registry:

```
docker login -u <REGISTRY_USERNAME> -p <REGISTRY_PASSWORD> <REGISTRY_NAME>


podman login -u <REGISTRY_USERNAME> -p <REGISTRY_PASSWORD> <REGISTRY_NAME>


# It will ask for password

# Enter the password generated while creating the auth token.
```

Where,

- `REGISTRY_NAME` is `<Region_Key>.ocir.io`
- `REGISTRY_USERNAME` is `<Object Storage Namespace>/<identity_domain>/email_id`
- `REGISTRY_PASSWORD` is the Authtocken generated by the user.

For the details about the Region Key, refer to Regions and Availability Domains.

Identity Domain will the domain,to which the user is present.

Object Storage Namespace is available at OCI Console> Governanace & Administration> Account Management> Tenenancy Details> Object Storage Namespace.

6. Run one of the following commands to tag each imported image to the registry:

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


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

7. Run one of the following commands to push the image to the registry:

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


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

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

8. Run the following command to push the helm charts to the helm repository:

```
helm push <image_name>.tgz <helm_repo>
```

9. Run the following command to extract the helm charts:

```
tar -xvzf ocnadd-25.2.200.tgz
```

10. Run the following command to unzip the custom_templates.zip file.

```
unzip custom_templates.zip
```

> **Note**
>
> All the image repositories must be public. Run the following steps to make all image repositories public:
>
> 1. Go to **OCI Console> Developer Services > Containers & Artifacts> Container Registry**.
> 2. Select the root Compartment.
> 3. In the **Repositories and Images Search** option, the images will be listed. Select each image and click **Change to Public**. This step must be performed for all the images sequentially.

2.2.1.3 Creating OCNADD Namespace

This section explains how to verify or create new namespaces in the system. In this section, the namespaces for the management group and worker group should be created.

Naming Convention for Namespaces

While choosing the name of the namespace where you wish to deploy OCNADD, make sure the following requirements are met:

- starts and ends with an alphanumeric character
- contains 63 characters or less
- contains only alphanumeric characters or '-'

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

Verifying Namespaces

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

```
kubectl get namespaces
```

If the namespace exists, you may continue with the next steps of installation.

If the required namespace is not available, create a namespace using the following command:

> **Note**
>
> The user must create the required namespaces for a centralized deployment with multiple worker groups. If the deployment mode is centralized with the default worker group, a single namespace is sufficient, and all Data Director services can be deployed within it

Creating Namespaces

Run the following command to create the namespace where OCNADD services will be deployed:

```
kubectl create namespace <ocnadd-group-namespace>
```

For Example:

```
# To create Management group namespace
kubectl create namespace ocnadd-mgmt
# To create Relay Agent group namespace
kubectl create namespace ocnadd-relay
# To create Mediation group namespace
kubectl create namespace ocnadd-med
```

Run the following command to verify the namespaces are created:

```
kubectl get namespaces
```

For example:

```
# kubectl get namepsaces
  ocnadd-mgmt
  ocnadd-rea
  ocnadd-med
```

2.2.1.4 Creating Service Account, Role, and Role Binding

This section is optional and it describes how to manually create a service account, role, and rolebinding. It is required only when customer needs to create a role, rolebinding, and service account manually before installing OCNADD. Skip this if choose to create by default from helm charts.

In the case of centralized deployment, this procedure needs to be repeated for each of the management group and worker group(s).

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

Creating Service Account, Role, and RoleBinding

To create the service account, role, and rolebinding:

1. **Prepare the Resource File:** Run the following command to create an OCNADD resource file:

   ```
   vi ocnadd-<ocnadd-group>-resource-file.yaml
   ```

   Replace `<ocnadd-group>` with the required group name.

   For example:

   ```
   vi ocnadd-management-resource-template.yaml
   ```

2. **Update the OCNADD Resource Template:** Update the created YAML file with release-specific information. A sample template to update the YAML file is given below:

   ```
   ## Sample template start #
   apiVersion: v1
   kind: ServiceAccount
   metadata:
     name: <namespace>-sa-ocnadd
     namespace: <namespace>
     automountServiceAccountToken: false

   ---
   apiVersion: rbac.authorization.k8s.io/v1
   kind: Role
   metadata:
     name: <namespace>-cr
   rules:
   - apiGroups: [""]
     resources: ["pods", "configmaps", "services", "secrets",
   "resourcequotas", "events", "persistentvolumes", "persistentvolumeclaims"]
     verbs: ["*"]
   - apiGroups: ["extensions"]
     resources: ["ingresses"]
     verbs: ["create", "get", "delete"]
   - apiGroups: [""]
     resources: ["nodes"]
     verbs: ["get"]
   - apiGroups: ["scheduling.volcano.sh"]
     resources: ["podgroups", "queues", "queues/status"]
     verbs: ["get", "list", "watch", "create", "delete", "update"]

   ---
   apiVersion: rbac.authorization.k8s.io/v1
   kind: RoleBinding
   metadata:
     name: <namespace>-crb
   roleRef:
     apiGroup: ""
     kind: Role
     name: <namespace>-cr
   ```

```
subjects:
- kind: ServiceAccount
  name: <namespace>-sa-ocnadd
  namespace: <namespace>

---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: <namespace>-crb-policy
roleRef:
  apiGroup: ""
  kind: ClusterRole
  name: psp:privileged
subjects:
- kind: ServiceAccount
  name: <namespace>-sa-ocnadd
  namespace: <namespace>

---
## Sample template end #
```

Replace `<namespace>` with the respective OCNADD group namespace.

3. **Create Service Account, Role, and RoleBinding:** Run the following command to create the service account, role, and rolebinding for the OCNADD group:

   ```
   kubectl -n <ocnadd-group-namespace> create -f ocnadd-<ocnadd-group>-
   resource-file.yaml
   ```

   Replace `<ocnadd-group-namespace>` with the namespace where the OCNADD group will be deployed.

   For example:

   ```
   $ kubectl -n ocnadd-mgmt create -f ocnadd-management-resource-template.yaml
   ```

2.2.1.5 Configuring OCNADD Database

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

The database is managed by the helm pre-install hook. However, OCNADD requires the database administrator to create an admin user in MySQL database and provide the necessary permissions to access the databases. Before installing OCNADD it is required to create the MySQL user and databases.

> **Note**
>
> - If the admin user is already available, then update the credentials, such as username and password (base64 encoded) in `<charts_directory>/templates/ocnadd-secret-hook.yaml.`.
> - If the admin user is not available, then create it using the following procedure. Once the user is created, update the credentials for the user in `<charts_directory>/templates/ocnadd-secret-hook.yaml.`.

Creating an Admin User in the Database

To create an admin user in the database:

1. Run the following command to access the MySQL pod. Use the namespace in which the `cnDBTier` is deployed.

   ```
   kubectl -n <cndbtier-namespace> exec -it <mysql-pod-name> -- bash
   ```

   Example: The `occne-cndbtier` namespace is used. The default MySQL pod name is `ndbmysqld-0`.

   ```
   kubectl -n occne-cndbtier exec -it ndbmysqld-0 -- bash
   ```

2. Run the following command to log in to the MySQL server using the MySQL client:

   ```
   $ mysql -h 127.0.0.1 -uroot -p
   Enter password:
   ```

3. To create an admin user, run the following command:

   ```
   CREATE USER IF NOT EXISTS '<ocnadd admin username>'@'%' IDENTIFIED BY
   '<ocnadd admin user password>';
   ```

   Example:

   ```
   CREATE USER IF NOT EXISTS 'ocdd'@'%' IDENTIFIED BY 'ocdd';
   ```

   Where:

   <ocdd> is the admin username and <ocdd> is the password for the MySQL admin user.

4. Run the following command to grant the necessary permissions to the admin user and run the FLUSH command to reload the grant table:

   ```
   GRANT ALL PRIVILEGES ON *.* TO 'ocdd'@'%' WITH GRANT OPTION;
   FLUSH PRIVILEGES;
   ```

5. Access the `ocnadd-secret-hook.yaml` from the OCNADD Helm files using the following path:

   ```
   ocnadd/templates/ocnadd-secret-hook.yaml
   ```

6. Update the following parameters in the `ocnadd-secret-hook.yaml` with the admin user credentials:

```
data:
  MYSQL_USER: b2NkZA==
  MYSQL_ACCESS_KEY: b2NkZA==
```

7. To generate the base64-encoded user and password from the terminal, run the following command:

```
echo -n <string> | base64 -w 0
```

Where `<string>` is the admin username or password created in step 3.

Example:

```
echo -n ocdd | base64 -w 0
b2NkZA==
```

Update Database Name

- **Default Database Names:**
  - `configuration_schema`
  - `alarm_schema`
  - `healthdb_schema`
  - `storageadapter_schema`

  These correspond to the Configuration Service, Alarm Service, and Health Monitoring Service respectively.
- **When to Update:**
  - If you plan to use the default database names, **skip this step**.
  - If you want to use custom database names, you must **modify them before installation**.
- **During Reinstallation:**
  Before reinstalling OCNADD, the four application databases **must be removed manually**.

  Run the following command for each database:

  ```
  drop database <dbname>;
  ```
- **Where to Update Database Names in Helm Charts:**
  To apply custom database names, update **all occurrences** of the required database names in the following file:

  ```
  <charts_directory>/charts/ocnaddmanagement/ocdd-db-resource.sql
  ```

2.2.1.6 Configuring Secrets for Accessing OCNADD Database

The secret configuration for OCNADD database is automatically managed during the database creation the helm preinstall procedure.

2.2.1.7 Configuring IP Network

This section defines OCNADD IP configuration for a single stack (either only IPv4 or IPv6) or a dual stack supported infrastructure.

> **Note**
>
> - The IP family remains fixed once OCNADD is deployed. To change the IP family, OCNADD must be redeployed.
> - IPv6 support on OCI is not available in the current release.

- For CNE with support for IPv4 or IPv6 networks:
  - **IPv4 only Configurations**: For the IPv4 network, update the following parameters in `ocnadd-common-custom-values.yaml`:

    ```
    global:
        ipConfigurations:
            ipFamilyPolicy: SingleStack
            ipFamilies: ["IPv4"]
    ```

  - **IPv6 only Configurations**: For the IPv6 network, update the following parameters in `ocnadd-common-custom-values.yaml`:

    ```
    global:
        ipConfigurations:
            ipFamilyPolicy: SingleStack
            ipFamilies: ["IPv6"]
    ```

2.2.1.8 Configuring SSL or TLS Certificates

In OCNADD, you can create SSL or TLS certificates using one of the following methods:

- Certificate generation using OCNADD script
- Certificate generation through Oracle Communication Certificate Manager (OCCM)

For step-by-step instructions on generating certificates, please refer to the *Oracle Communications Network Analytics Suite Security Guide* in the "Certificate and Secret Generation" section.

2.2.1.9 Configuring ServiceMonitor in OCCNE-INFRA

This section defines OCCNE-INFRA OCNADD ServiceMonitor configuration to scrape Kafka Prometheus metrics.

ocnadd-servicemonitorfile

```
cd ocnadd-package-25.2.200
sudo cp custom_templates/occne_ocnadd_servicemonitor.yaml <path>/
occne_ocnadd_servicemonitor.yaml
```

Log in as root or any user with admin privileges and execute the following command to apply:

```
kubectl -n occne-infra apply -f <path>/occne_ocnadd_servicemonitor.yaml
```

2.2.2 Installation Tasks

This section describes the tasks that the user must follow for installing OCNADD.

> **Note**
>
> Before starting the installation tasks, ensure that the Prerequisites and Pre-Installation Tasks are completed.

2.2.2.1 Installing OCNADD Package

This section describes how to install the Oracle Communications Network Analytics Data Director (OCNADD) package. OCNADD deployment now has three groups, that is, management, relayagent, and mediation.

Refer to the following steps to install different OCNADD groups:

Installing OCNADD Management Group

1. **Create OCNADD Namespace**: Create the OCNADD namespace for the Management Group, if not already created. For more information, see Creating OCNADD Namespace.
2. **Creating Service Account, Role, and Role Binding**: If the user has opted to manually create a service account, role, and role binding, follow the steps outlined in the Creating Service Account, Role, and Role Binding section.
3. **Generate Certificates**: Follow the steps outlined in the Configuring SSL or TLS Certificates section to complete the certificate generation process for the Management Group if it is not performed.
4. **Update Database Parameters**: To update the database parameters, see Configuring OCNADD Database.
5. **Update Custom Values file**: Create a copy of the custom values for the Management Group from the `ocnadd-package-25.2.200` folder as shown below:

   ```
   # cd ocnadd-package-25.2.200
   # cp custom_templates/ocnadd-management-custom-values-25.2.200.yaml ocnadd-management-custom-values-25.2.200-mgmt-group.yaml
   ```

   Update the `ocnadd-common-custom-values-25.2.200.yaml` and `ocnadd-management-custom-values-25.2.200-mgmt-group.yaml` (depending on the type of deployment model) with the required parameters. For more information on how to access and update the custom values files, see Customizing OCNADD.

   If OCCM is used to create the certificates, update the mandatory parameters specified in Helm Parameter Configuration for OCCM.
6. **Installing Management Group**:

a. Modify the `ocnadd-management-custom-values-25.2.200-mgmt-group.yaml` file created above and update it as shown below:

```
`global.ocnaddmanagement.cluster.namespace.name`: ocnadd-mgmt
        ## ---> update it with namespace created for management group

`global.ocnaddmanagement.cluster.serviceAccount.create`: true
        ## ---> update this to false only if service account is created
manually

`global.ocnaddmanagement.cluster.clusterRole.create`: true
        ## ---> update this to false only if cluster role is created
manually

`global.ocnaddmanagement.cluster.clusterRoleBinding.create`: true
        ## ---> update this to false only if cluster role binding is
created manually

`global.ocnaddmanagement.cluster.serviceAccount.name`: ocnadd
        ## ---> update ocnadd with namespace created for management
group

`global.ocnaddmanagement.cluster.clusterRole.name`: ocnadd
        ## ---> update ocnadd with namespace created for management
group

`global.ocnaddmanagement.cluster.clusterRoleBinding.name`: ocnadd
        ## ---> update ocnadd with namespace created for management
group
```

b. Install the Management Group using the OCNADD Helm charts folder:

```
helm install <management-release-name> -f ocnadd-common-custom-
values-25.2.200.yaml -f <management-custom-values> --namespace
<management-namespace> <helm_chart>
```

where:

`<management-release-name>` release name of Management Group deployment

`<management-custom-values>` Management custom values file

`<management-namespace>` namespace where Management Group is deployed

`<helm-chart>` Helm chart folder of OCNADD

Example:

```
helm install OCNADD-mgmt -f ocnadd-common-custom-values-25.2.200.yaml -
f ocnadd-management-custom-values-25.2.200-mgmt-group.yaml --namespace
ocnadd-mgmt ocnadd
```

Installing OCNADD RelayAgent Group

1. **Create OCNADD Namespace**: Create the OCNADD namespace for the Relay Agent Group, if not already created. For more information, see Creating OCNADD Namespace.

2. **Creating Service Account, Role, and Role Binding**: If the user has opted to manually create a service account, role, and role binding, follow the steps outlined in the Creating Service Account, Role, and Role Binding section.
3. **Generate Certificates**: Follow the steps outlined in the Configuring SSL or TLS Certificates section to complete the certificate generation process for the Relay Agent Group if it is not performed.
4. **Update Custom Values file**: Create a copy of the custom values for the Relay Agent Group from the `ocnadd-package-25.2.200` folder as shown below:

   ```
   # cd ocnadd-package-25.2.200
   # cp custom_templates/ocnadd-relayagent-custom-values-25.2.200.yaml ocnadd-relayagent-custom-values-25.2.200-ra-group.yaml
   ```

   Update the `ocnadd-common-custom-values-25.2.200.yaml` and `ocnadd-relayagent-custom-values-25.2.200-ra-group.yaml` (depending on the type of deployment model) with the required parameters. For more information on how to access and update the custom values files, see Customizing OCNADD.

   If OCCM is used to create the certificates, update the mandatory parameters specified in Helm Parameter Configuration for OCCM.
5. **Install Relay Agent Group**:
   a. Modify `ocnadd-common-custom-values-25.2.200.yaml` and update it as shown below:

      ```
      `global.management_info.management_namespace`: ocnadd-management
              ## ---> update it with namespace created for management group
      ```

   b. Modify the `ocnadd-relayagent-custom-values-25.2.200-ra-group.yaml` file created above and update it as shown below:

      ```
      `global.ocnaddrelayagent.cluster.namespace.name`: ocnadd-relay
              ## ---> update it with namespace created for relayagent group

      `global.ocnaddrelayagent.cluster.serviceAccount.create`: true
              ## ---> update this to false only if service account is created manually

      `global.ocnaddrelayagent.cluster.clusterRole.create`: true
              ## ---> update this to false only if cluster role is created manually

      `global.ocnaddrelayagent.cluster.clusterRoleBinding.create`: true
              ## ---> update this to false only if cluster role binding is created manually

      `global.ocnaddrelayagent.cluster.serviceAccount.name`: ocnadd
              ## ---> update ocnadd with namespace created for relayagent group

      `global.ocnaddrelayagent.cluster.clusterRole.name`: ocnadd
              ## ---> update ocnadd with namespace created for relayagent group

      `global.ocnaddrelayagent.cluster.clusterRoleBinding.name`: ocnadd
      ```

```
        ## ---> update ocnadd with namespace created for relayagent
group
```

c. Install the Relay Agent Group using the OCNADD Helm charts folder:

```
helm install <relayagent-release-name> -f ocnadd-common-custom-
values-25.2.200.yaml -f <relayagent-custom-values> --namespace
<relayagent-namespace> <helm_chart>
```

where:

`<relayagent-release-name>` release name of Relay Agent Group deployment

`<relayagent-custom-values>` Relay Agent custom values file

`<relayagent-namespace>` namespace where Relay Agent Group is deployed

`<helm-chart>` Helm chart folder of OCNADD

Example:

```
helm install OCNADD-ra -f ocnadd-common-custom-values-25.2.200.yaml -f
ocnadd-relayagent-custom-values-25.2.200-ra-group.yaml --namespace
ocnadd-relay ocnadd
```

Installing OCNADD Mediation Group

1. **Create OCNADD Namespace**: Create the OCNADD namespace for the Mediation Group, if not already created. For more information, see Creating OCNADD Namespace.
2. **Creating Service Account, Role, and Role Binding**: If the user has opted to manually create a service account, role, and role binding, follow the steps outlined in the Creating Service Account, Role, and Role Binding section.
3. **Generate Certificates**: Follow the steps outlined in the Configuring SSL or TLS Certificates section to complete the certificate generation process for the Mediation Group if it is not performed.
4. **Update Custom Values file**: Create a copy of the custom values for the Mediation Group from the `ocnadd-package-25.2.200` folder as shown below:

   ```
   # cd ocnadd-package-25.2.200
   # cp custom_templates/ocnadd-mediation-custom-values-25.2.200.yaml ocnadd-
   mediation-custom-values-25.2.200-med-group.yaml
   ```

   Update the `ocnadd-common-custom-values-25.2.200.yaml` and `ocnadd-mediation-custom-values-25.2.200-med-group.yaml` (depending on the type of deployment model) with the required parameters. For more information on how to access and update the custom values files, see Customizing OCNADD.

   If OCCM is used to create the certificates, update the mandatory parameters specified in Helm Parameter Configuration for OCCM.
5. **Installing Mediation Group**:

a. Modify the `ocnadd-mediation-custom-values-25.2.200-med-group.yaml` file created above and update it as shown below:

```
`global.ocnaddmediation.cluster.namespace.name`: ocnadd-deploy
        ## ---> update it with namespace created for mediation group

`global.ocnaddmediation.cluster.serviceAccount.create`: true
        ## ---> update this to false only if service account is created
manually

`global.ocnaddmediation.cluster.clusterRole.create`: true
        ## ---> update this to false only if cluster role is created
manually

`global.ocnaddmediation.cluster.clusterRoleBinding.create`: true
        ## ---> update this to false only if cluster role binding is
created manually

`global.ocnaddmediation.cluster.serviceAccount.name`: ocnadd
        ## ---> update ocnadd with namespace created for mediation group

`global.ocnaddmediation.cluster.clusterRole.name`: ocnadd
        ## ---> update ocnadd with namespace created for mediation group

`global.ocnaddmediation.cluster.clusterRoleBinding.name`: ocnadd
        ## ---> update ocnadd with namespace created for mediation group
```

b. Install the Mediation Group using the OCNADD Helm charts folder:

```
helm install <mediation-release-name> -f ocnadd-common-custom-
values-25.2.200.yaml -f <mediation-custom-values> --namespace
<mediation-namespace> <helm_chart>
```

where:

`<mediation-release-name>` release name of Mediation Group deployment

`<mediation-custom-values>` Mediation custom values file

`<mediation-namespace>` namespace where Mediation Group is deployed

`<helm-chart>` Helm chart folder of OCNADD

Example:

```
helm install OCNADD-med -f ocnadd-common-custom-values-25.2.200.yaml -f
ocnadd-mediation-custom-values-25.2.200-med-group.yaml --namespace
ocnadd-med ocnadd
```

2.2.2.2 Verifying OCNADD Installation

This section describes how to verify if *Oracle Communications Network Analytics Data Director Disaster Recovery Guide* (OCNADD) is installed successfully.

To check the status of the OCNADD deployment, perform the following task:

1. Run the following commands to check Helm release status:

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

   Example:

   To check release status for the management group:

   ```
   # helm status dd-mgmt -n ocnadd-mgmt
   ```

   To check release status for the relay agent group:

   ```
   # helm status dd-ra -n ocnadd-relay
   ```

   To check release status for the mediation group:

   ```
   # helm status dd-med -n ocnadd-med
   ```

   The system displays the status as *deployed* if the deployment is successful.
2. Run the following command to check whether all the services are deployed and active:

   ```
   watch kubectl get pod,svc -n <ocnadd-group-namespace>
   ```

   Example:

   To check the status of pods for the management group:

   ```
   # watch kubectl get pod,svc -n ocnadd-mgmt
   ```

   To check the status of pods for the relay agent group:

   ```
   # watch kubectl get pod,svc -n ocnadd-relay
   ```

   To check the status of pods for the mediation group:

   ```
   # watch kubectl get pod,svc -n ocnadd-med
   ```

> **Note**
>
> - All microservices status must be *Running* and *Ready*.
> - Take a backup of the following files that are required during fault recovery:
>   - Updated Helm charts for Management, Relay Agent, and Mediation Group(s)
>   - Updated custom values for Management, Relay Agent, and Mediation Group(s)
>   - Secrets, certificates, and keys that are used during the installation for Management, Relay Agent, and Mediation Group(s)
> - If the installation is not successful or you do not see the status as *Running* for all pods, perform the troubleshooting steps. For more information, refer to the *Oracle Communications Network Analytics Data Director Troubleshooting Guide.*.

2.2.2.3 Creating OCNADD Kafka Topics

To create OCNADD Kakfa topics, see the "Creating Kafka Topic for OCNADD" section of *Oracle Communications Network Analytics Data Director User Guide*

2.2.2.4 Installing OCNADD GUI

This section describes how to install Oracle Communications Network Analytics Data Director (OCNADD) GUI using the following steps:

- Install OCNADD GUI
- Configure OCNADD GUI in CNC Console
- Access OCNADD GUI

Install OCNADD GUI

The OCNADD GUI gets installed along with the OCNADD services.

Configure OCNADD GUI in CNCC

Prerequisite: To configure OCNADD GUI in CNC Console, you must have the CNC Console installed. For information on how to install CNC Console and configure the OCNADD instance, see *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

Before installing CNC Console, ensure to update the instances parameters with the following details in the `occncc_custom_values.yaml` file:

```
instances:
   - id: Cluster1-dd-instance1
     type: DD-UI
     owner: Cluster1
     ip: 10.xx.xx.xx      #--> give the cluster/node IP
     port: 31456          #--> give the node port of ocnaddgui
     apiPrefix: /<clustername>/<namespace>/ocnadd
   - id: Cluster1-dd-instance1
     type: DD-API
     owner: Cluster1
```

```
        ip: 10.xx.xx.xx      #--> give the cluster/node IP
        port: 32406          #--> give the node port of ocnaddbackendrouter
        apiPrefix: /<clustername>/<namespace>/ocnaddapi

# Applicable only for Manager and Agent core. Used for Multi-Instance-Multi-
Cluster Configuration Validation
  validationHook:
     enabled: false #--> add this enabled: false to validationHook

#--> do these changes under section :
cncc iam attributes
# If https is disabled, this Port would be HTTPS/1.0 Port (secured SSL)
publicHttpSignalingPort: 30085 #--> CNC console nodeport

#--> add these lines under cncc-iam attributes
# If Static node port needs to be set, then set staticNodePortEnabled flag to
true and provide value for staticNodePort
     # Else random node port will be assigned by K8
     staticNodePortEnabled: true
     staticHttpNodePort: 30085 #--> CNC console nodeport
     staticHttpsNodePort: 30053

#--> do these changes under section : manager cncc core attributes
#--> add these lines under mcncc-core attributes

# If Static node port needs to be set, then set staticNodePortEnabled flag to
true and provide value for staticNodePort
    # Else random node port will be assigned by K8
    staticNodePortEnabled: true
    staticHttpNodePort: 30075 staticHttpsNodePort: 30043

#--> do these changes under section : agent cncc core attributes
#--> add these lines under acncc-core attributes
# If Static node port needs to be set, then set staticNodePortEnabled flag to
true and provide value for staticNodePort
     # Else random node port will be assigned by K8
     staticNodePortEnabled: true
     staticHttpNodePort: 30076
     staticHttpsNodePort: 30044
```

If CNC Console is already installed, ensure to upgrade it with the following parameters updated in the `occncc_custom_values.yaml` file:

```
instances:
   - id: Cluster1-dd-instance1
     type: DD-UI owner: Cluster1
     ip: 10.xx.xx.xx         #--> update the cluster/node IP
     port: 31456             #--> ocnaddgui port
     apiPrefix: /<clustername>/<management_group_namespace>/ocnadd
   - id: Cluster1-dd-instance1
     type: DD-API owner: Cluster1
     ip: 10.xx.xx.xx         #--> update the cluster/node IP
     port: 32406             #--> ocnadduirouter port
     apiPrefix: /<clustername>/<management_group_namespace>/ocnaddapi
```

Example:

If OCNADD GUI is deployed in the **occne-ocdd** cluster and the **ocnadd-mgmt,** namespace, then the prefix in CNC Console `occncc_custom_values.yaml` will be as follows:

```
DD-UI apiPrefix:
/occne-ocdd/ocnadd-mgmt/ocnadd
DD-API apiPrefix:
/occne-ocdd/ocnadd-mgmt/ocnaddapi
```

Access OCNADD GUI

To access OCNADD GUI, follow the procedure mentioned in the "Accessing CNC Console" section of *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

2.2.2.5 Adding a Mediation Group

> **Note**
>
> Adding a Mediation Group is possible using the steps listed below; however, the user must note that all possible scenarios for this feature have not been verified, and you may encounter a few challenges while installing an additional Mediation Group.

Assumptions:

- OCNADD is already deployed with a worker group consisting of a Relay Agent and at least one Mediation Group.
- Management Group deployment is up and running (for example, in namespace *ocnadd-mgmt*).
- To utilize the extended storage feature with MySQL in the Mediation Group being deployed to a new cluster, ensure that network connectivity is established between the new cluster and the MySQL cluster hosting the management group.

Procedure for Adding a New Mediation Group:

1. Follow the section Installing OCNADD Mediation Group to customize and install another Mediation Group in the same worker group.
2. To verify the installation of the new Mediation Group, run the following command:

   ```
   # watch kubectl get pod,svc -n ocnadd-med2
   ```

3. Follow the section Creating OCNADD Kafka Topics to create topics on the newly added Mediation Group.
4. Once a Mediation Group is registered with the Management Group, all existing feed configurations are automatically migrated to the newly added Mediation Group. If any issues arise during this feed replication process, you can reinitiate the replication using the steps outlined below:

a. **Access the configuration service pod in the Management Group namespace**

```
kubectl exec -ti -n <management-group-namespace> <configuration-service-pod-name> -- bash
```

b. **Run the following `curl` command to re-trigger the feed replication**

```
curl -k -X GET --location "http://ocnaddmanagementgateway:12889/ocnadd-configuration/v1/deploy-resources/<mediation-group-name>"
```

c. **If secure communication (mTLS: true) is enabled, use:**

```
curl -k -X GET --location --cert-type P12 --cert /var/securityfiles/keystore/serverKeyStore.p12:$OCNADD_SERVER_KS_PASSWORD "https://ocnaddmanagementgateway:12889/ocnadd-configuration/v1/deploy-resources/<mediation-group-name>"
```

Where:

`<mediation-group-name>` = `<SiteName>:<WorkerName>:<Namespace>:<ClusterName>`

Example

If the following values are configured:

```
siteName: BLR              ## common custom values
workergroupName: wg1       ## mediation custom values
clusterName: cluster-1     ## mediation custom values
```

And the Mediation Group is deployed in the `ocnadd-med2` namespace, then the Mediation Group name will be:

```
BLR:wg1:ocnadd-med2:cluster-1
```

Example command:

```
curl -k -X GET --location --cert-type P12 --cert /var/securityfiles/keystore/serverKeyStore.p12:$OCNADD_SERVER_KS_PASSWORD "https://ocnaddmanagementgateway:12889/ocnadd-configuration/v1/deploy-resources/BLR:wg1:ocnadd-med2:cluster-1"
```

d. **Verify completion**: Verify that the feed replication has been successfully completed. Note that this process may take some time.

2.2.2.6 Deleting a Mediation Group

Assumptions:

- OCNADD is already deployed with at least one Mediation Group.
- Management Group deployment is up and running, for example, namespace "ocnadd-mgmt".
- Mediation groups "mediation-group-1" and "mediation-group-2" deployment is up and running, for example, namespace "ocnadd-med1" and "ocnadd-med2".
- The Mediation group "mediation-group-2" needs to be deleted.

1. **Clean up the configurations corresponding to the mediation group being deleted.** *(Skip this step when changing the Kafka storage mode.)*
   a. Delete all the adapter feeds corresponding to mediation-group-2 using the curl command.
   b. Delete all the filters applied to mediation-group-2 using the curl command.
   c. Delete all the correlations applied to mediation-group-2 using the curl command.
   d. Delete all the Kafka feeds corresponding to mediation-group-2 using the curl command.

```
curl -k -X DELETE --location "http://ocnaddmanagementgateway:12889/
ocnadd-configuration/v1/delete-resources/<mediation-group-name>"
```

If secure communication for DD is enabled (mTLS: true):

```
curl -k -X DELETE --location --cert-type P12 \
--cert /var/securityfiles/keystore/
serverKeyStore.p12:$OCNADD_SERVER_KS_PASSWORD \
"https://ocnaddmanagementgateway:12889/ocnadd-configuration/v1/delete-
resources/<mediation-group-name>"
```

where

`<mediation-group-name>` = `<SiteName>:<WorkerName>:<Namespace>:<ClusterName>`

Example:

Given the following parameter values and the mediation group deployed in ocnadd-med2 namespace:

```
siteName: BLR
workergroupName: wg1
clusterName: cluster-1
```

Then the mediation group name will be:

```
BLR:wg1:ocnadd-med2:cluster-1
```

The command will be:

```
curl -k -X DELETE --location --cert-type P12 \
--cert /var/securityfiles/keystore/
serverKeyStore.p12:$OCNADD_SERVER_KS_PASSWORD \
"https://ocnaddmanagementgateway:12889/ocnadd-configuration/v1/delete-
resources/BLR:wg1:ocnadd-med2:cluster-1"
```

If some feeds from the mediation group are not deleted due to a network failure, re-execute the deletion step to ensure all feeds are removed.

> **Note**
>
> For the scenario where only one Mediation Group is deployed, clean up the configurations corresponding to the Mediation Group being deleted using the OCNADD UI:
>
> - Delete all the adapter feeds corresponding to the worker group.
> - Delete all the filters applied to the worker group.
> - Delete all the correlations applied to the worker group.
> - Delete all the Kafka feeds corresponding to the worker group.

2. **Uninstall the Mediation Group:**

   ```
   helm uninstall <mediation-group2-release-name> -n <mediation-group2-namespace>
   ```

   Example:

   ```
   helm uninstall dd-med2 -n ocnadd-med2
   ```

3. **Delete the Mediation Group namespace.**
   (Skip this step when changing the Kafka storage mode.)

   ```
   kubectl delete namespace <mediation-group2-namespace>
   ```

   Example:

   ```
   kubectl delete namespace ocnadd-med2
   ```

2.2.2.7 Deleting a Relay Agent Group

Assumptions:

- OCNADD is already deployed with at least one Mediation Group.
- Management Group deployment is up and running, for example, namespace "ocnadd-mgmt".
- Relay Agent group "relayagent-group-1" deployment is up and running, for example, namespace "ocnadd-rea1".
- The Relay Agent group "relayagent-group-1" needs to be deleted.

1. **Clean up the configurations corresponding to the Relay Agent group being deleted.**
   (Skip this step when changing the Kafka storage mode.)
   - Delete the Global OCNADD Metadata configuration from the UI.
   - Delete all the Ingress filters applied to this Relay Agent group from the UI.
2. **Uninstall the Relay Agent group:**

   ```
   helm uninstall <relayagent-group1-release-name> -n <relayagent-group1-namespace>
   ```

Example:

```
helm uninstall dd-rea1 -n ocnadd-rea
```

3. **Delete the Relay Agent group namespace.**
   (Skip this step when changing the Kafka storage mode.)

   ```
   kubectl delete namespace <relayagent-group1-namespace>
   ```

   Example:

   ```
   kubectl delete namespace ocnadd-rea
   ```

2.2.2.8 Deleting a Worker Group

Deletion of a worker group involves the removal of all sub-groups, including relay agent groups and mediation groups, that are associated with it.

Assumptions

- The OCNADD is already deployed with at least one worker group.
- Management Group deployment is up and running, for example, namespace "ocnadd-mgmt".
- Worker group "worker-group1" has Relay Agent "relayagent-group1" and "mediation-group1" and their deployment is up and running, for example, namespace "ocnadd-rea" and "ocnadd-med".
- The worker group "worker-group1" needs to be removed. Therefore, the associated Relay Agent "relayagent-group1" and Mediation "mediation-group1" group must be removed.

1. **Clean up the configurations corresponding to the worker group being deleted.**
   For example, if it is "worker-group1":
   - Delete all the adapter feeds corresponding to worker-group1 from the UI.
   - Delete all the filters applied to worker-group1 from the UI.
   - Delete all the correlations applied to worker-group1 from the UI.
   - Delete all the Kafka feeds corresponding to worker-group1 from the UI.
2. **Uninstall the Mediation group:**

   ```
   helm uninstall <mediation-group1-release-name> -n <mediation-group1-
   namespace>
   ```

   Example:

   ```
   helm uninstall dd-med1 -n ocnadd-med
   ```

3. **Delete the Mediation group namespace:**

   ```
   kubectl delete namespace <mediation-group1-namespace>
   ```

Example:

```
kubectl delete namespace ocnadd-med
```

4. **Uninstall the Relay Agent group:**

```
helm uninstall <relayagent-group1-release-name> -n <relayagent-group1-
namespace>
```

Example:

```
helm uninstall dd-rea1 -n ocnadd-rea
```

5. **Delete the Relay Agent group namespace:**

```
kubectl delete namespace <relayagent-group1-namespace>
```

Example:

```
kubectl delete namespace ocnadd-rea
```

2.2.2.9 Creating Alarms and Dashboard in OCI

This step is necessary only for the Data Director deployment on the OCI platform. Follow the steps explained in the section 'Creating Alarms and Dashboards in OCI' from the *Oracle Communications Network Analytics Data Director User Guide*.

2.2.2.10 Adding or Updating Load Balancer IPs in SAN When OCCM is Used

The certificates created by OCCM will not contain any IP values in the SAN field except the values provided in the `global.<groupname>.certificate.occm.san.*.ips` field in the custom values file for Kafka broker, ingress adapter, redundancy agent, and gateway certificates.

To add or update the LoadBalancer IPs of these services in the SAN, follow the steps mentioned below. Refer to Helm Parameter Configuration for OCCM for a detailed description of the different Helm parameters.

2.2.2.10.1 Adding Loadbalancer IPs for Management Gateway Services

Adding LoadBalancer IPs for Management Gateway Services

1. Update the `global.ocnaddmanagement.certificates.occm.san.management_gateway.ips` field in `ocnadd-management-custom-values-25.2.200.yaml` of the required management group.
   Update Management Gateway LoadBalancer IPs

```
global:
  ocnaddmanagement:
    certificates:
      occm:
        san:
          management_gateway:
```

```
            ips: ["10.10.10.10"]     # Add the LoadBalancer IP of the
management gateway service
```

2. a. If single certificate is **not** enabled, update the `global.ocnaddmanagement.certificates.occm.san.management_gateway.update_required` and `global.ocnaddmanagement.certificates.occm.san.management_gateway.uuid.server` fields in `ocnadd-management-custom-values-25.2.200.yaml` of the required management group.
   Management Gateway SAN upgrade

```
global:
  ocnaddmanagement:
    certificates:
      occm:
        san:
          management_gateway:
            update_required: true          # Set to true, default is
false
            uuid:
              server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e     #
Provide the UUID value of the certificate MANAGEMENTGATEWAY-SECRET-
SERVER-<namespace> from OCCM, where <namespace> is the management group
namespace
```

   b. If single certificate **is enabled** for OCNADD, update the `global.certificates.occm.san.ocnadd.update_required` and `global.certificates.occm.san.ocnadd.uuid.server` fields in `ocnadd-common-custom-values-25.2.200.yaml`.
   OCNADD SAN upgrade

```
global:
  certificates:
    occm:
      san:
        ocnadd:
          update_required: true          # Set to true, default is false
          uuid:
            server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e     # Provide
the UUID value of the certificate OCNADD-SECRET-SERVER-<namespace> from
OCCM, where <namespace> is the mediation group namespace
```

3. Run Helm upgrade for the Management Group namespace.
   Helm upgrade

```
helm upgrade <management-group-release-name> -f <common-custom-values> -f
<management-group-custom-values> -n <management-group-ns> <ocnadd-helm-
chart-location>
```

4. New certificates will be created. Verify them through the OCCM UI. Management Gateway will also restart after the Helm upgrade is completed and will start using the newly created certificates.

2.2.2.10.2 Adding LoadBalancer IP for Redundancy Agent

1. Update the `global.ocnaddmanagement.certificates.occm.san.redundancy_agent.ips` field in `ocnadd-management-custom-values-25.2.200.yaml` of the required management group.
   Update Agent LoadBalancer IPs

```
global:
  ocnaddmanagement:
    certificates:
      occm:
        san:
          redundancy_agent:
            ips: ["10.10.10.10"]     # Add the LoadBalancer IP of the
redundancy agent service
```

2. a. If single certificate is **not** enabled, update the `global.ocnaddmanagement.certificates.occm.san.redundancy_agent.update_required` and `global.ocnaddmanagement.certificates.occm.san.redundancy_agent.uuid.server` fields in `ocnadd-management-custom-values-25.2.200.yaml` of the required management group.
   Redundancy Agent SAN upgrade

```
global:
  ocnaddmanagement:
    certificates:
      occm:
        san:
          redundancy_agent:
            update_required: true          # Set to true, default is
false
            uuid:
              server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e     #
Provide the UUID value of the certificate REDUNDANCYAGENT-SECRET-SERVER-
<namespace> from OCCM, where <namespace> is the management group
namespace
```

   b. If single certificate **is enabled** for OCNADD, update the `global.certificates.occm.san.ocnadd.update_required` and `global.certificates.occm.san.ocnadd.uuid.server` fields in `ocnadd-common-custom-values-25.2.200.yaml`.
   OCNADD SAN upgrade

```
global:
  certificates:
    occm:
      san:
        ocnadd:
          update_required: true          # Set to true, default is false
          uuid:
            server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e     # Provide
the UUID value of the certificate OCNADD-SECRET-SERVER-<namespace> from
OCCM, where <namespace> is the mediation group namespace
```

3. Run Helm upgrade for the Management Group namespace.
Helm upgrade

```
helm upgrade <management-group-release-name> -f <common-custom-values> -f
<management-group-custom-values> -n <management-group-ns> <ocnadd-helm-
chart-location>
```

4. New certificates will be created. Verify them through the OCCM UI. The Redundancy Agent will restart after the Helm upgrade and begin using the newly created certificates.

2.2.2.10.3 Adding Loadbalancer IPs for Relay Agent Kafka

1. Update the `global.ocnaddrelayagent.certificates.occm.san.kafka.ips` field in `ocnadd-relayagent-custom-values-25.2.200.yaml` of the required Relay Agent group.
Update Relay Agent Kafka LoadBalancer IPs

```
global:
  ocnaddrelayagent:
    certificates:
      occm:
        san:
          kafka:
            ips: ["10.10.10.10", "10.10.10.11", "10.10.10.12",
"10.10.10.13"]   # Add the LoadBalancer IP of each Kafka broker service
```

2. a. If single certificate is **not** enabled, update `global.ocnaddrelayagent.certificates.occm.san.kafka.update_required` and `global.ocnaddrelayagent.certificates.occm.san.kafka.uuid.server` in the same custom values file.
Relay Agent Kafka SAN upgrade

```
global:
  ocnaddrelayagent:
    certificates:
      occm:
        san:
          kafka:
            update_required: true        # Set to true, default is false
            uuid:
              server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
KAFKABROKER-SECRET-SERVER-<namespace> from OCCM
```

b. If single certificate **is enabled** for OCNADD, update `global.certificates.occm.san.ocnadd.update_required` and `global.certificates.occm.san.ocnadd.uuid.server` in `ocnadd-common-custom-values-25.2.200.yaml`.
OCNADD SAN upgrade

```
global:
  certificates:
    occm:
      san:
        ocnadd:
          update_required: true         # Set to true, default is false
          uuid:
```

```
                server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
OCNADD-SECRET-SERVER-<namespace> from OCCM
```

3. Run Helm upgrade for the Relay Agent group namespace.
   Helm upgrade

   ```
   helm upgrade <relayagent-release-name> -f <common-custom-values> -f
   <relayagent-group-custom-values> -n <relayagent-group-ns> <ocnadd-helm-
   chart-location>
   ```

4. New certificates will be created. Verify them through the OCCM UI. Kafka brokers will restart after the Helm upgrade and will start using the newly created certificates.

2.2.2.10.4 Adding LoadBalancer IPs for Relay Agent Gateway Services

1. Update the `global.ocnaddrelayagent.certificates.occm.san.management_gateway.ips` field in `ocnadd-relayagent-custom-values-25.2.200.yaml` of the required Relay Agent group.
   Update Relay Agent Gateway LoadBalancer IPs

   ```
   global:
     ocnaddrelayagent:
       certificates:
         occm:
           san:
             relay_gateway:
               ips: ["10.10.10.10"]    # Add the LoadBalancer IP of the Relay
   Agent Gateway service
   ```

2. a. If single certificate is **not** enabled, update `global.ocnaddrelayagent.certificates.occm.san.relay_gateway.update_required` and `global.ocnaddrelayagent.certificates.occm.san.relay_gateway.uuid.server` in the same custom values file.
   Relay Agent Gateway SAN upgrade

   ```
   global:
     ocnaddrelayagent:
       certificates:
         occm:
           san:
             relay_gateway:
               update_required: true       # Set to true, default is false
               uuid:
                 server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
   RELAYGATEWAY-SECRET-SERVER-<namespace> from OCCM
   ```

   b. If single certificate **is enabled** for OCNADD, update `global.certificates.occm.san.ocnadd.update_required` and `global.certificates.occm.san.ocnadd.uuid.server` in `ocnadd-common-custom-values-25.2.200.yaml`.
   OCNADD SAN upgrade

   ```
   global:
     certificates:
   ```

```
    occm:
      san:
        ocnadd:
          update_required: true         # Set to true, default is false
          uuid:
            server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
OCNADD-SECRET-SERVER-<namespace> from OCCM
```

3. Run Helm upgrade for the Relay Agent group namespace.
   Helm upgrade

   ```
   helm upgrade <relayagent-group-release-name> -f <common-custom-values> -f
   <relayagent-group-custom-values> -n <relayagent-group-ns> <ocnadd-helm-
   chart-location>
   ```

4. New certificates will be created; verify them in the OCCM UI. The Relay Gateway will restart after the Helm upgrade is completed and will start using the newly created certificates.

2.2.2.10.5 Adding LoadBalancer IPs for Mediation Kafka

1. Update the `global.ocnaddmediation.certificates.occm.san.kafka.ips` field in `ocnadd-mediation-custom-values-25.2.200.yaml` of the required Mediation group.
   Update Mediation Kafka LoadBalancer IPs

   ```
   global:
     ocnaddmediation:
       certificates:
         occm:
           san:
             kafka:
               ips: ["10.10.10.10", "10.10.10.11", "10.10.10.12",
   "10.10.10.13"]   # Add the LoadBalancer IPs of each Kafka broker service
   ```

2. a. If single certificate is **not** enabled, update `global.ocnaddmediation.certificates.occm.san.kafka.update_required` and `global.ocnaddmediation.certificates.occm.san.kafka.uuid.server` in the same custom values file.
   Mediation Kafka SAN upgrade

   ```
   global:
     ocnaddmediation:
       certificates:
         occm:
           san:
             kafka:
               update_required: true       # Set to true, default is false
               uuid:
                 server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
   KAFKABROKER-SECRET-SERVER-<namespace> from OCCM
   ```

   b. If single certificate **is enabled** for OCNADD, update `global.certificates.occm.san.ocnadd.update_required` and `global.certificates.occm.san.ocnadd.uuid.server` in `ocnadd-common-custom-values-25.2.200.yaml`.

OCNADD SAN upgrade

```
global:
  certificates:
    occm:
      san:
        ocnadd:
          update_required: true         # Set to true, default is false
          uuid:
            server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
OCNADD-SECRET-SERVER-<namespace> from OCCM
```

3. Run Helm upgrade for the Mediation group namespace.
   Helm upgrade

   ```
   helm upgrade <mediation-release-name> -f <common-custom-values> -f
   <mediation-group-custom-values> -n <mediation-group-ns> <ocnadd-helm-chart-
   location>
   ```

4. New certificates will be created; verify them through the OCCM UI. Kafka Brokers will restart after the Helm upgrade is completed and will begin using the newly generated certificates.

2.2.2.10.6 Adding LoadBalancer IPs for Ingress Adapter

1. Update the `global.ocnaddmediation.certificates.occm.san.ingress_adapter.ips` field in `ocnadd-mediation-custom-values-25.2.200.yaml` of the required Mediation group.
   Update Ingress Adapter LoadBalancer IPs

   ```
   global:
     ocnaddmediation:
       certificates:
         occm:
           san:
             ingress_adapter:
               ips: ["10.10.10.10", "10.10.10.11", "10.10.10.12",
   "10.10.10.13"]   # Add the LoadBalancer IPs of each Ingress Adapter service
   ```

2. a. If single certificate is **not enabled**, update `global.ocnaddmediation.certificates.occm.san.ingress_adapter.update_required` and `global.ocnaddmediation.certificates.occm.san.ingress_adapter.uuid.server` in the same custom values file.
   Ingress Adapter SAN upgrade

   ```
   global:
     ocnaddmediation:
       certificates:
         occm:
           san:
             ingress_adapter:
               update_required: true      # Set to true, default is false
               uuid:
   ```

```
                server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
INGRESSADAPTER-SECRET-SERVER-<namespace> from OCCM
```

b. If single certificate **is enabled**, update`global.certificates.occm.san.ocnadd.update_required` and`global.certificates.occm.san.ocnadd.uuid.server` in `ocnadd-common-custom-values-25.2.200.yaml`.
OCNADD SAN upgrade

```
global:
  certificates:
    occm:
      san:
        ocnadd:
          update_required: true        # Set to true, default is false
          uuid:
            server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e   # UUID of
OCNADD-SECRET-SERVER-<namespace> from OCCM
```

3. Run the Helm upgrade for the Mediation group namespace.
Helm upgrade

```
helm upgrade <mediation-group-release-name> -f <common-custom-values> -f
<mediation-group-custom-values> -n <mediation-group-ns> <ocnadd-helm-chart-
location>
```

4. New certificates will be created with the updated SAN entries. Verify them through the OCCM UI.
5. Run a second Helm upgrade to apply the updated certificates after restart.

```
helm upgrade <mediation-group-release-name> -f <common-custom-values> -f
<mediation-group-custom-values> -n <mediation-group-ns> <ocnadd-helm-chart-
location> --set
global.ocnaddmediation.env.admin.OCNADD_INGRESS_ADAPTER_UPGRADE_ENABLE=true
```

2.2.2.10.7 Adding Loadbalancer IPs for Mediation Gateway Services

1. Update the `global.ocnaddmediation.certificates.occm.san.management_gateway.ips` in `ocnadd-mediation-custom-values-25.2.200.yaml` of the required Mediation group.
Update Mediation Gateway Load balancer IPs

```
global:
  ocnaddmediation:
    certificates:
      occm:
        san:
          mediation_gateway:
            ips: ["10.10.10.10"]              # Add the loadbalancer IP of
the mediation gateway service
```

2. a. If single certificate is not enabled, then update the `global.ocnaddmediation.certificates.occm.san.mediation_gateway.update_required` and

`global.ocnaddmediation.certificates.occm.san.mediation_gateway.uuid.server` in `ocnadd-mediation-custom-values-25.2.200.yaml` of the required Mediation group.
Mediation Gateway SAN upgrade

```
global:
  ocnaddmediation:
    certificates:
      occm:
        san:
          mediation_gateway:
            update_required: true                                    #
Set to true, default is false
            uuid:
              server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e           #
Provide the UUID value of the certificate MEDIATIONGATEWAY-SECRET-
SERVER-<namespace> from OCCM, where <namespace> is the Mediation group
namespace
```

b. If single certificate is enabled for OCNADD, then update the `global.certificates.occm.san.ocnadd.update_required` and `global.certificates.occm.san.ocnadd.uuid.server` in `ocnadd-common-custom-values-25.2.200.yaml`.
OCNADD SAN upgrade

```
global:
  certificates:
    occm:
      san:
        ocnadd:
          update_required: true                                  # Set
to true, default is false
          uuid:
            server: 5e765aeb-ae1b-426b-8481-f8f3dcdd645e         #
Provide the UUID value of the certificate OCNADD-SECRET-SERVER-
<namespace> from OCCM, where <namespace> is the Mediation group
namespace
```

3. **Run Helm upgrade for the Mediation group namespace.**

```
$ helm upgrade <mediation-group-release-name> -f <common-custom-values> -f
<mediation-group-custom-values> -n <mediation-group-ns> <ocnadd-helm-chart-
location>
```

4. New certificates will be created; verify the same through the OCCM UI. Mediation Gateway will also restart after the Helm upgrade is completed and will start using the newly created certificates.

2.2.3 Post-Installation Tasks

2.2.3.1 Enabling Traffic Segregation Using CNLB

This feature is introduced as part of traffic segregation support in OCNADD. To enable it, see 'Enabling or Disabling Traffic Segregation Using CNLB in OCNADD ' section in the *Oracle*

Communications Network Analytics Data Director User Guide. It is recommended to enable this feature after completing the deployment of the target release.

2.2.3.2 Enabling Two Site Redundancy

This feature is introduced as part of Georedundancy in OCNADD. To enable it, see 'Two Site Redundancy Enable' section in the *Oracle Communications Network Analytics Data Director User Guide*. It is recommended to enable this feature after completing the deployment of the target release.

2.2.3.3 Enabling Druid as Extended Storage Feature

This feature is introduced as part of extended storage in Data Director. To enable it, see the *"Druid Integration with OCNADD"* section in the *Oracle Communications Network Analytics Data Director User Guide*. The feature is recommended to be enabled after the release installation is completed. Extended storage using the cnDBTier database is available by default.

2.2.3.4 vCollector Integration for Diameter Feed

In this release, integration with vCollector has been provided. The vCollector acquires the Diameter traffic from vDSR using port mirroring. The vCollector is deployed as a virtual machine outside the OCNADD cluster and provides the acquired Diameter traffic to Data Director over the Kafka interface. The vCollector is configured and managed by the Data Director OAM services. This feature is introduced as part of the Diameter feed capabilities in Data Director. To enable integration with vCollector, refer to the *vCollector Integration with Data Director* section in the *Oracle Communications Network Analytics Data Director Diameter User Guide*. The feature is recommended to be enabled after the release installation is completed.

3
Customizing OCNADD

This chapter describes how to customize the Oracle Communications Network Analytics Data Director (OCNADD) deployment, supported deployment models, and provides a list of configuration parameters in the Helm file that are used for customization. The OCNADD deployment is customized by overriding the default values of various configurable parameters.

3.1 Customize Configuration Parameters

Perform the following procedure to customize the OCNADD custom values files as per requirements for all the OCNADD groups.

- Ensure that you have the Data Director charts tgz file, which is available in the extracted release package. For information about how to download the release package from MOS, see *Downloading OCNADD Package*.
- Extract the OCNADD package if not already extracted, and unzip the `custom_templates.zip`. Change the directory to `custom_templates` to access the following custom values file
  - `ocnadd-common-custom-values-25.2.200.yaml`
  - `ocnadd-management-custom-values-25.2.200.yaml`
  - `ocnadd-relayagent-custom-values-25.2.200.yaml`
  - `ocnadd-mediation-custom-values-25.2.200.yaml`

3.1.1 Modify the Commons custom values file

Edit the `ocnadd-common-custom-values-25.2.200.yaml` and update the following parameters:

1. Update the repository path in `global.env.repo.REPO_HOST_PORT`:

   ```
   <customer
              repository path>
   ```

2. Update the site name in `cluster.siteName` (for example Bangalore, Chicago, etc.).
3. Update the database IP and database name:
   - `db_ip`: `10.20.30.40` (Update with DB instance IP or with FQDN, for example `mysql-connectivity-service.<cnDBTier namespace>`.)
   - `db_port`: `3306` (If using a different port for DB, change it. By default, the DB port is 3306.)
   - `configuration_db`: `configuration_schema` (Update the DB name as per the section *Update Database Name*. No change is needed if default DB names are used.)
   - `alarm_db`: `alarm_schema` (Update the DB name as per the section *Update Database Name*. No change is needed if default DB names are used.)
   - `health_db`: `healthdb_schema` (Update the DB name as per the section *Update Database Name*. No change is needed if default DB names are used.)

4. Change the Prometheus Monitoring details based on the desired MPS profile. The default threshold MPS is 1500000:

   ```
   cluster.mps: 1500000
   ```

5. <Optional>Update JAAS Password for Mediation Kafka
   The following JAAS password property is used for authentication between Aggregation Services and the Mediation Kafka Cluster. Update the following property to change the JAAS password:

   `cluster.aggJaasSecret`: `<jaas password>`

6. Creating Registry Credentials:
   If the user image repository is protected and has an authentication mechanism, follow the steps below:

   a. Use the `kubectl` command to create a secret named `regcred` with the credentials of the image repository.

   b. Update the `imagePullSecret.enable` field to `true` as follows:

      ```
      imagePullSecret:
        enable: true    ## --> update this to 'true'
      ```

7. Storage class:

   a. If deploying on Tanzu, update the `storageClass` with the respective storage class name of the TANZU platform, for example `zfs-storage-policy`.

   b. If deploying on OCI, update the `storageClass` with the respective storage class name of the OCI platform. It should be `oci-bv`.

   > **Note**
   >
   > This step is specific to the TANZU and OCI platform. Skip this step if you are installing OCNADD on CNE. For CNE, the default storageClass is standard.

8. Loadbalancer on OCI:
   Update the following parameters in the file:

   ```
   global.env.oci: false   =====> set it to true
   global.env.subnetOcid:<subnet_ocid>   # Add the OCID of the subnet that
   user want to use for creating load balancer
   ```

3.1.2 Modifying Management custom values file

Edit the `ocnadd-management-custom-values-25.2.200.yaml` and update the following parameters:

1. Update the `CLUSTER-INFO` parameters:

   ```
   cluster.domainName: <cluster domain name where management group is planned
   for deployment>
   cluster.clusterName: <cluster name where management group is planned for
   deployment>
   ```

2. <Optional> Updating the OCNADD Backup Cronjob:
   Modify the below backup parameters as needed. For more information on backup and restore, see the "Fault Recovery" section.

```
BACKUP_STORAGE : Represents Backup storage PVC size
BACKUP_CRONEXPRESSION : Represents the time of execution
PURGE_DAYS : Represents the backup retention period in days

ocnaddbackuprestore:
   ocnaddbackuprestore:
        name: ocnaddbackuprestore
        env:
             BACKUP_STORAGE: 20Gi
             BACKUP_CRONEXPRESSION: "0 8 * * *"
             PURGE_DAYS: 7
```

3. <Optional> When the management group is deployed in a separate cluster from the relay agent or mediation group, external communication between gateways must be enabled. For detailed instructions on enabling external communication, refer to "Enable External Communication Between OCNADD Gateways" in the *Oracle Communications Network Analytics Data Director User Guide*.

3.1.3 Modifying Relay Agent custom values file

Update the management namespace in commons custom values (`ocnadd-common-custom-values-25.2.200.yaml`) file if it is not updated.

```
global.management_info.management_namespace: <management group namespace>
```

If the relay agent group is deployed in a separate cluster from the management group, then external communication between the gateways must be enabled. For detailed instructions on enabling external communication, refer to *Enable External Communication Between OCNADD Gateways* in the *Oracle Communications Network Analytics Data Director User Guide*.

Edit the relay agent YAML file (for example, `ocnadd-relayagent-custom-values-25.2.200-ra-group.yaml`) created after making a copy of the `ocnadd-relayagent-custom-values-25.2.200.yaml` and update the following parameters:

1. Update the CLUSTER-INFO parameters:

```
cluster.domainName: <cluster domain name where relay agent group is
planned for deployment>
cluster.clusterName: <cluster name where relay agent group is planned for
deployment>
```

2. Update the logical name of the worker group to which this relay agent group is associated:

```
cluster.workergroupName: <Logical name of the worker group> (for example
worker1 or myddworker)
```

3. Enable the specific Aggregation service. The default values are mentioned below. The user can enable the specific Aggregation service by setting `true` against the specific Aggregation service:

```
global:
    ocnaddrelayagent:
        ocnaddscpaggregation:
            enabled: true
        ocnaddseppaggregation:
            enabled: false
        ocnaddnrfaggregation:
            enabled: false
        ocnaddbsfaggregation:
            enabled: false
        ocnaddpcfaggregation:
            enabled: false
```

4. Kafka Preinstall Configuration Changes:
   a. <Optional> To change the profiles of the brokers, edit the respective values (CPU, memory, storage, external-access, security, `jaas-password`, replicas, internal replication factor, and so on) in the Kafka section of the Relay Agent custom values file.
   b. <Optional> When the security protocol is SASL and the customer is required to add new users, update the `kafka_server_jaas.conf` files in `<chartpath>/ocnadd/charts/ocnaddrelayagent/charts/ocnaddkafka/config`.
   c. <Optional> Customize `offsetsTopicReplicationFactor` and `transactionStateLogReplicationFactor` under `ocnaddkafka.ocnadd.kafkaBroker.kafkaProperties` in the Relay Agent custom values file.
   d. Update Internal Topic Replication Factor:

```
# Update to below values when higher throughput with lower latency is
needed.
# This can have lower message reliability in case a Kafka broker goes
down.
offsetsTopicReplicationFactor: 1
transactionStateLogReplicationFactor: 1

# Update to below values when higher message reliability is required
(RF>1).
# This can potentially have lower throughput and higher latency if the
Kafka cluster Disk IOPS and cluster network bandwidth are less
performing.
offsetsTopicReplicationFactor: 2
transactionStateLogReplicationFactor: 2
```

3.1.4 Modifying Mediation custom values file

Update the management namespace in commons custom values (`ocnadd-common-custom-values-25.2.200.yaml`) file if it is not updated.

```
global.management_info.management_namespace: <management group namespace>
```

If the mediation group is deployed in a separate cluster from the management group, then external communication between the gateways must be enabled. For detailed instructions on enabling external communication, refer to *Enable External Communication Between OCNADD Gateways* in the *Oracle Communications Network Analytics Data Director User Guide*.

Edit the Mediation YAML file (for example, `ocnadd-mediation-custom-values-25.2.200-med-group1.yaml`) created after making a copy of the `ocnadd-mediation-custom-values-25.2.200.yaml` and update the following parameters:

1. Update the CLUSTER-INFO parameters:

   ```
   cluster.domainName: <cluster domain name where mediation group is planned
   for deployment>
   cluster.clusterName: <cluster name where mediation group is planned for
   deployment>
   ```

2. Update the logical name of the worker group to which the mediation group is associated. (Avoid using a colon in the name.)

   ```
   cluster.workergroupName: <Logical name of the worker group> (for example
   worker1 or myddworker)
   ```

3. Update the associated relay agent namespace cluster information to which the mediation group is associated:

   ```
   global.ocnaddmediation.deployment.associatedRelayAgentNsClusterInfo:
   <Relay agent namespace cluster information>
   ```

   Example:

   If the Mediation Group you want to associate with a Relay Agent Group is deployed in the "ocnadd-deploy" namespace on "cluster-1", update the configuration parameter as follows:

   ```
   global.ocnaddmediation.deployment.associatedRelayAgentNsClusterInfo:
   "ocnadd-deploy:cluster-1"
   ```

4. Kafka Preinstall Configuration Changes:

   a. <Optional> To change the profiles of the brokers, edit the respective values (CPU, memory, storage, external-access, security, `jaas-password`, replicas, internal replication factor, and so on) in the Kafka section of the Mediation custom values file.

   b. <Optional> When the security protocol is SASL and the customer is required to add new users, update the `kafka_server_jaas.conf` files in `<chartpath>/ocnadd/charts/ocnaddmediation/charts/ocnaddkafka/config`.

   c. <Optional> Customize `offsetsTopicReplicationFactor` and `transactionStateLogReplicationFactor` under `ocnaddkafka.ocnadd.kafkaBroker.kafkaProperties` in the Mediation custom values file.
   Update Internal Topic Replication Factor:

   ```
   # Update to below values when higher throughput with lower latency is
   needed.
   # This can have lower message reliability in case a Kafka broker goes
   down.
   offsetsTopicReplicationFactor: 1
   ```

```
transactionStateLogReplicationFactor: 1

# Update to below values when higher message reliability is required
(RF>1).
# This can potentially have lower throughput and higher latency if the
Kafka cluster Disk IOPS and cluster network bandwidth are less
performing.
offsetsTopicReplicationFactor: 2
transactionStateLogReplicationFactor: 2
```

d. <Optional> Enable RAM-based storage for the Kafka cluster. This feature has been introduced to support RAM-based storage in the Kafka cluster. It supports higher throughput for cases where lower message retention is needed with lower latency. To enable RAM-based storage in the Kafka cluster, refer to *Enable RAM Storage in Kafka Cluster* section in the *Oracle Communications Network Analytics Data Director User Guide*.

3.1.5 OCNADD UI Configurations Changes for Dashboard Metrics

Edit the `ocnadd-management-custom-values-25.2.200.yaml` and update the `DD_GROUP_PROMETHEUS_API` for UI with Relay Agent and Mediation group names and Prometheus IP & Port to enable metrics from all OCNADD groups in the UI dashboard:

```
## The parameter is a comma-separated list of group name (Relay Agent or
Mediation) and Prometheus IP and Port
## Add all Relay Agent and Mediation groups and their corresponding
Prometheus IP & Port

ocnadduirouter:
    ocnadduirouter:
        name: ocnadduirouter
        env:
            groupNamePromIpConfig:
                - groupName: site1:worker1:ocnadd-deploy1:cluster-1
                  prometheusIp: http://localhost:9099
                - groupName: site2:worker2:ocnadd-deploy2:cluster-2
                  prometheusIp: http://localhost:9098
```

For example:

1. When OCNADD UI is deployed at the 'BLR' site, where the Relay Agent and Mediation groups are co-located within cluster-1, share the same worker group (`wg1`), and are deployed under namespaces `"dd-relay-agent"` & `"dd-mediation"` respectively, the parameter should be updated as follows:

   ```
   groupNamePromIpConfig:
       - groupName: BLR:wg1:dd-relay-agent:cluster-1
         prometheusIp: http://localhost:9099
       - groupName: BLR:wg1:dd-mediation:cluster-1
         prometheusIp: http://localhost:9099
   ```

2. When OCNADD UI is deployed at the 'BLR' site, where the Relay Agent and Mediation groups exist in different clusters (for example, cluster-1 and cluster-2 respectively), share

the same worker group (wg1), and are deployed under namespaces "dd-relay-agent" & "dd-mediation" respectively, the parameter should be updated as follows:

```
groupNamePromIpConfig:
    - groupName: BLR:wg1:dd-relay-agent:cluster-1
      prometheusIp: http://localhost:9099
    - groupName: BLR:wg1:dd-mediation:cluster-2
      prometheusIp: http://localhost:9100
```

3. When OCNADD UI is deployed at the 'BLR' site, where the Relay Agent and multiple Mediation groups exist in different clusters (for example, Relay Agent: cluster-1, Mediation1: cluster-1, and Mediation2: cluster-2), share the same worker group (wg1), and are deployed under namespaces "dd-relay-agent", "dd-mediation1", and "dd-mediation2" respectively, the parameter should be updated as follows:

```
groupNamePromIpConfig:
    - groupName: BLR:wg1:dd-relay-agent:cluster-1
      prometheusIp: http://localhost:9099
    - groupName: BLR:wg1:dd-mediation1:cluster-1
      prometheusIp: http://localhost:9099
    - groupName: BLR:wg1:dd-mediation2:cluster-2
      prometheusIp: http://localhost:9100
```

3.1.6 Alerting Rules Configuration Updates

1. If OCNADD is to be installed in an OCI setup, remove the following files:
   - <chartpath>/charts/ocnaddmanagement/templates/ocnadd-mgmt-alerting-rules.yaml
   - <chartpath>/charts/ocnaddrelayagent/templates/ocnadd-relayagent-alerting-rules.yaml
   - <chartpath>/charts/ocnaddmediation/templates/ocnadd-mediation-alerting-rules.yaml
2. If OCNADD is to be installed in a CNE setup, all services will be monitored by Prometheus by default. Therefore, there will not be any modifications in the Helm Chart. All the Prometheus alert rules present in the Helm Chart will be updated in the Prometheus server. (The label used to update the Prometheus server is role: cnc-alerting-rules, which is added by default in the Helm Charts.)
3. If OCNADD is to be installed in a Tanzu setup, modify the metadata.labels value in the following files as described:
   - <chartpath>/charts/ocnaddmanagement/templates/ocnadd-mgmt-alerting-rules.yaml
   - <chartpath>/charts/ocnaddrelayagent/templates/ocnadd-relayagent-alerting-rules.yaml
   - <chartpath>/charts/ocnaddmediation/templates/ocnadd-mediation-alerting-rules.yaml

   For example, use release: prom-operator" instead of "role: cnc-alerting-rules.

To obtain the label details, use the following command:

```
kubectl get prometheus <Prometheus_Configuration_NAME> -n
<Prometheus_Namespace> -o=jsonpath='{.spec.ruleSelector.matchLabels}'
```

Example:

```
kubectl get prometheus prom-operator-kube-prometh-prometheus -n occne-
infra -o=jsonpath='{.spec.ruleSelector.matchLabels}'{"release: prom-
operator"}
```

Sample Alert File:

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
    labels:
      release: prom-operator
    name: ocnadd-alerting-rules
    namespace: {{ .Values.global.cluster.nameSpace.name }}
...
...
```

3.2 Global Parameters

Table 3-1 Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
Common Custom Values Global Parameters					
management_info.management_gateway_ip	STRING	NA	-	C	This parameter should be enabled when external communication is required across OCNADD groups. The parameter denotes the gateway IP of management group. This is required only while installing relay agent group or mediation group.
management_info.management_namespace	STRING	NA	ocnadd-management	M	The management group namespace name required while installing relay agent or mediation groups
ocnaddredundancyagent.enabled	BOOLEAN	true/ false	false	M	To enable two site redundancy charts
ocnaddredundancyagent.egress	BOOLEAN	true/ false	false	C	Required if egress annotation is required to allow traffic outside cluster
ocnaddredundancyagent.primary_site	BOOLEAN	true/ false	false	M	True if upgrading from non-centralized to centralized.Default is False (direct installation)

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddredundancyagent.primary_agent_ip	STRING	NA	-	C	This parameter is required only in case the redundancy agent service is enabled. It is configured in the secondary site and denotes the primary site redundancy agent IP address or service fqdn.
extendedStorage.druid.enabled	BOOLEAN	true/ false	false	M	Enable if Druid database as extended storage is available, else cnDBTier as extended storage is used by default
extendedStorage.druid.druidTLsEnabled	BOOLEAN	true/ false	true	O	The parameter depicts if TLS should be used for communication with Druid services. Default is true
extendedStorage.druid.namespace	String	NA	ocnadd-druid	M	The namespace in which Druid cluster is deployed, if deployed in the same cluster as Data Director
extendedStorage.druid.service_ip	String	NA	1.1.1.1	M	The loadbalancer of the Druid router service, this must only be changed if Druid is enabled else leave as is
extendedStorage.druid.service_port	String	NA	8080	M	The port of the Druid router service
extendedStorage.druid.secret_name	String	NA	ocnadd-druid-api-secret	M	The name of the secret containing the Druid API credentials.
certificates.singlecert	BOOLEAN	true/ false	false	M	Enable this parameter only when single certificate is created (using OCCM or generate certificate script) for each OCNADD group
initContainers.repo.REPO_PATH	STRING	NA	utils.repo	M	Repo path where init image is stored
initContainers.cacert.value	STRING	NA	cacert.pem	M	Cacert file
cnlb.consumeradapter.enable	BOOLEAN	true/ false	false	C	If true, then egress traffic segregation will be enabled for the consumer feeds. It will be used for checking if Egress NAD should be attached to consumer adapters or not. The parameter is only recommended to be enabled for the OCCNE with CNLB support
cnlb.ingressadapter.enable	BOOLEAN	true/ false	false	C	If true, then ingress traffic segregation will be enabled for the ingress adapter along with external access. It will be used for checking if Ingress NAD should be attached to ingress adapters or not. The parameter is only recommended to be enabled for the OCCNE with CNLB support

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
env.repo.REPO_HOST_PORT	STRING	NA	docker.io	M	Local container registry to pull the images
env.repo.REPO_PATH	STRING	NA	ocdd.repo	M	Additional repo path
env.oci	BOOLEAN	true/ false	false	M	It should be set to true for OCI platform
env.subnetOcid	STRING	NA	NA	C	It is required for OCI platform. The OCID of the subnet that you want to use for creating load balancers
env.topologyKey	STRING	NA	kubernetes.io/ hostname	M	The topology key for setting up the topology constraints on the pod deployment
stabilizationWindowSeconds	INTEGER	NA	60	M	Stabilization period in seconds post which scale down starts
scaleDownPeriodSeconds	INTEGER	NA	30	M	Period of each scale down operation in seconds
scaleDownValue	INTEGER	NA	1	M	Number of pods which shall go down in every scaleDownPeriodSeconds
controlPlaneNfList	STRING	NA	BSF,NRF,PCF	M	It enlists all the control plane NFs
proxyNfList	STRING	NA	SCP,SEPP	M	It enlist all the proxy NFs
ssl.mTLS	BOOLEAN	true/ false	false	M	Enable MTLS support for internal OCNADD services

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ssl.kafkaCipherSuites	STRING	NA	"TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"	M	Supported Cipher Suites for Kafka Broker service in Data Director

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ssl.tlsCipherSuites	STRING	NA	"TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"	M	Support Cipher Suites for Internal Services of Data Director
ssl.CERT_EXPIRY_CRONEXPRESSION	STRING	NA	0 0 * * *	M	Schedule for Cron Expression that will check certificate expiry at midnight everyday
ssl.CERT_EXPIRY_CRONJOB	BOOLEAN	true/ false	true	M	Enable cronjob schedule to check certificate expiry
acl.genericAclAllowed	BOOLEAN	true/ false	false	M	No need to change this flag here, genericAclAllowed=true will be used in upgrade --set command to restrict the generic ACL creation.
acl.kafkaClientAuth	STRING	none/ required	none	M	This Property is to enable or disable MTLS in Kafka.
acl.aclNotAllowed	BOOLEAN	true/ false	true	M	This Property is used to turn on or off the Kafka ACL's.

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
cluster.terminationGracePeriodSeconds	INTEGER	NA	5	O	Pod grace termination
cluster.siteName	STRING	NA	BLR	M	The site name where OCNADD is deployed (e.g. Chicago, BLR)
cluster.secret.name	STRING	NA	db-secret	M	Database Secret name where DB credentials are stored
cluster.mysqlNameSpace.name	STRING	NA	occne-cndbtierone	M	cnDBTier namespace
cluster.mysqlPod	STRING	NA	ndbmysqld-0	M	cnDBTier Pod Name
cluster.database.db_ip	STRING	NA	mysql-connectivity-service.occne-cndbtierone`1	M	Hostname or IP of cnDBTier
cluster.database.db_port	INTEGER	NA	3306	M	DB Port
cluster.database.configuration_db	STRING	NA	configuration_schema	M	Configuration Service Schema Name
cluster.database.alarm_db	STRING	NA	alarm_schema	M	Alarm Service Schema Name
cluster.database.health_db	STRING	NA	healthdb_schema	M	Health Service Schema Name
cluster.database.storageadapter_db	STRING	NA	storageadapter_schema	M	Storage Adapter Schema Name
cluster.storageClass	STRING	NA	standard	M	Storage Class Name
cluster.prometheusScrapePort	INTEGER	NA	9000	O	Port to scape metrics required if metrics enabled
cluster.prometheusPortName	STRING	NA	cnc-metrics	O	Role required to define in alert rules yaml
cluster.max_latency	FLOAT	NA	0.10	M	Max latency range of 100ms
cluster.memory_threshold	INTEGER	[0-100]	90	M	Max Threshold limit for memory
cluster.cpu_threshold	INTEGER	[0-100]	85	M	CPU max threshold limit
cluster.mps	INTEGER	NA	1500000	M	Default MPS rate

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
cluster.serviceMonitorLabel	STRING	NA	occne-kube-prom-stack	M	service monitor label to scrape metrics
Management Custom Values Global Parameters					
ocnaddmanagement.ocnaddalarm.enabled	BOOLEAN	true/ false	true	M	To enable alarm charts
ocnaddmanagement.ocnaddconfiguration.enabled	BOOLEAN	true/ false	true	M	To enable configuration charts
ocnaddmanagement.ocnaddhealthmonitoring.enabled	BOOLEAN	true/ false	true	M	To enable health monitoring charts
ocnaddmanagement.ocnaddbackuprestore.enabled	BOOLEAN	true/ false	true	M	To enable backup restore charts
ocnaddmanagement.ocnadduirouter.enabled	BOOLEAN	true/ false	true	M	To enable UI router charts
ocnaddmanagement.ocnaddgui.enabled	BOOLEAN	true/ false	true	M	To enable GUI charts
ocnaddmanagement.ocnaddexport.enabled	BOOLEAN	true/ false	false	M	To enable export service charts
ocnaddmanagement.ocnaddmanagementgateway.enabled	BOOLEAN	true/ false	true	M	To enable Management gateway charts
ocnaddmanagement.cnlb.ocnaddredundancyagent.enable	BOOLEAN	true/ false	false	C	If true, External Access will be enabled for RedundancyAgent. It will be used for checking if cnlb annotations should be assigned to the redundancy deployments or not. The parameter is only recommended to be enabled for the OCCNE with CNLB support Default is false

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmanagement.cnlb.ocnaddredundancyagent.network	STRING	NA	default/nf-oam-int7@nf-oam-int7	C	This must be the oam network with ingress definition. The entry means single network will be used by another site's Redundancy Agent for ingress communication. Update the network from the generated cnlb annotation, Given value is an example need to be updated as per /var/occne/cluster/$OCCNE_CLUSTER/artifacts/cnlbGenAnnotations.py script.
ocnaddmanagement.cnlb.ocnaddredundancyagent.externalIP	STRING	NA	-	C	For two site RedundancyAgent communication only single External IP required to communicate. Update the IP address in the External IP
ocnaddmanagement.cnlb.ocnaddmanagementgateway.enable	BOOLEAN	true/ false	false	C	If true, External Access will be enabled for Management Gateway. It will be used for checking if cnlb annotations should be assigned to the Management gateway deployments or not. The parameter is only recommended to be enabled for the OCCNE with CNLB support
ocnaddmanagement.cnlb.ocnaddmanagementgateway.network	STRING	NA	default/nf-oam-ie1@nf-oam-ie1	C	This must be the oam network with ingress-egress definition. The entry means single network will be used by gateways present in other sites for communication. Update the network from the generated cnlb annotation, given value is an example need to be updated as per /var/occne/cluster/$OCCNE_CLUSTER/artifacts/cnlbGenAnnotations.py script.
ocnaddmanagement.cnlb.ocnaddmanagementgateway.externalIP	STRING	NA	-	C	Single External IP is required to establish communication between gateways of different OCNADD groups located in other sites. Update the IP address in the External IP
ocnaddmanagement.cnlb.ocnadduirouter.enable	BOOLEAN	true/ false	false	C	If true, External Access will be enabled for UI router. It will be used for checking if cnlb annotations should be assigned to the UI Router deployments or not. The parameter is only recommended to be enabled for the OCCNE with CNLB support

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmanagement.cnlb.ocnadduirouter.network	STRING	NA	default/nf-oam-egr1@nf-oam-egr1	C	Single External IP is required to route traffic outside cluster Update the IP address in the External IP
ocnaddmanagement.migration.enable	BOOLEAN	true/ false	false	C	Enable this parameter while performing migration of configurations from source release to target release.
ocnaddmanagement.migration.sourceNamespace	STRING	NA	old_namespace	C	This parameter denotes the source release namespace from where the configuration should be migrated to the target release.
ocnaddmanagement.cluster.domainName	STRING	NA	occne-ocnadd	M	Domain name of the setup where Management Group will be deployed
ocnaddmanagement.cluster.clusterName	STRING	NA	occne-ocnadd	M	Cluster name of the setup where Management Group will be deployed
ocnaddmanagement.cluster.nameSpace.name	STRING	NA	ocnadd-management	M	OCNADD Management Group namespace
ocnaddmanagement.cluster.env.ALARM_PURGE_DAYS	INTEGER	NA	7	M	Alarm Purge in Days
ocnaddmanagement.cluster.env.OCNADD_MAX_EXTERNAL_KAFKA_FEEDS	INTEGER	NA	3	M	Maximum number of allowed external Kafka Feed
Relay Agent Custom Values Global Parameters					
ocnaddrelayagent.ocnaddscpaggregation.enabled	BOOLEAN	true/ false	true	M	To enable SCP aggregation charts
ocnaddrelayagent.ocnaddnrfaggregation.enabled	BOOLEAN	true/ false	false	M	To enable NRF aggregation charts
ocnaddrelayagent.ocnaddseppaggregation.enabled	BOOLEAN	true/ false	false	M	To enable SEPP aggregation charts
ocnaddrelayagent.ocnaddbsfaggregation.enabled	BOOLEAN	true/ false	false	M	To enable BSF aggregation charts
ocnaddrelayagent.ocnaddpcfaggregation.enabled	BOOLEAN	true/ false	false	M	To enable PCF aggregation charts

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddrelayagent.ocnaddvcollectordiameteraggregation	BOOLEAN	true/ false	false	M	To enable vCollector Diameter Aggregation charts
ocnaddrelayagent.ocnaddkafka.enabled	BOOLEAN	true/ false	true	M	To enable Relay Agent Kafka charts
ocnaddrelayagent.ocnaddrelayagentgateway.enabled	BOOLEAN	true/ false	true	M	To enable Relay Agent gateway charts
ocnaddrelayagent.deployment.gwToMgmtGwExternalCommEnabled	BOOLEAN	true/ false	false	C	Enable this property if Relay agent group is deployed in a separate cluster and external connectivity is required between management gateway and relay agent gateway. The communication between the gateways will be in secure mode.
ocnaddrelayagent.deployment.relayAgentGatewayIp	STRING	NA	-	C	This property is for configuring Load Balancer IP for Relay Agent gateway when external connectivity is enabled. Add the IP Addresses in the SAN entry as well for SSL communication This parameter is not applicable when cluster is CNLB enabled. For CNLB enabled setup, refer cnlb section to configure external IPs.
ocnaddrelayagent.cnlb.kafkabroker.enable	BOOLEAN	true/ false	false	C	If true, external access for the Relay Agent Kafka brokers will be enabled on the CNLB enabled cluster
ocnaddrelayagent.cnlb.kafkabroker.networks	STRING	NA	"default/nf-sig2-int1@nf-sig2-int1,default/nf-sig2-int2@nf-sig2-int2,default/nf-sig2-int3@nf-sig2-int3,default/nf-sig2-int4@nf-sig2-int4"	C	This property is used for populating the annotation k8s.v1.cni.cncf.io/networks Update the networks and networks_extip with the details generated by the section "Enable CNLB for Kafka Broker" from "OCNADD User Guide"

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddrelayagent.cnlb.kafkabroker.networks_extip	STRING	NA	"nf-sig2-int1/10.10.10.10, nf-sig2-int2/10.10.10.11, nf-sig2-int3/10.10.10.12, nf-sig2-int4/10.10.10.13"	C	This property is used for populating the annotation oracle.com.cnc/cnlb Add the IP Addresses in the SAN entry as well for SSL communication
ocnaddrelayagent.kafka.kafkaReplicas	INTEGER	NA	4	M	The parameter to change the replicas for the Relay Agent Kafka broker.
ocnaddrelayagent.cluster.domainName	STRING	NA	occne-ocdd	M	Domain name of the setup where Relay Agent Group will be deployed
ocnaddrelayagent.cluster.clusterName	STRING	NA	occne-ocdd	M	Cluster name of the setup where Relay Agent Group will be deployed
ocnaddrelayagent.cluster.workergroupName	STRING	NA	wg1	M	The logical name of the worker group to which this relay agent group belongs.
ocnaddrelayagent.cluster.nameSpace.name	STRING	NA	ocnadd-relay	M	OCNADD Relay Agent Group namespace
ocnaddrelayagent.cluster.kafka.ocnadd_kafka_bootstrap_servers	STRING	NA	kafka-broker:9092	M	Relay Agent Kafka Bootstrap server for PLAINTEXT
ocnaddrelayagent.cluster.kafka.ocnadd_kafka_bootstrap_servers_ssl	STRING	NA	kafka-broker:9093	M	Relay Agent Kafka Bootstrap server for SSL
ocnaddrelayagent.cluster.kafka.ocnadd_kafka_bootstrap_servers_sasl	STRING	NA	kafka-broker:9094	M	Relay Agent Kafka Bootstrap server for SASL
Mediation Custom Values Global Parameters					
ocnaddmediation.ocnaddkafka.enabled	BOOLEAN	true/ false	true	M	To enable Mediation Kafka charts
ocnaddmediation.ocnaddadmin.enabled	BOOLEAN	true/ false	true	M	To enable Admin charts
ocnaddmediation.ocnaddfilter.enabled	BOOLEAN	true/ false	false	M	To enable Filter charts

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmediation.ocnaddmediationgateway.enabled	BOOLEAN	true/ false	true	M	To enable Mediation gateway charts
ocnaddmediation.deployment.gwToMgmtGwExternalCommEnabled	BOOLEAN	true/ false	false	C	Enable this property if Mediation group is deployed in a separate cluster and external connectivity is required between management gateway and mediation gateway. The communication between the gateways will be in secure mode.
ocnaddmediation.deployment.mediationGatewayIp	STRING	NA	-	C	This property is for configuring Load Balancer IP for Mediation gateway when external connectivity is enabled. Add the IP Addresses in the SAN entry as well for SSL communication This parameter is not applicable when cluster is CNLB enabled. For CNLB enabled setup, refer cnlb section to configure external IPs.
ocnaddmediation.deployment.associatedRelayAgentNsClusterInfo	STRING	NA	xxx	M	Specify the namespace and cluster name of the Relay Agent that you want to associate with the mediation group.
ocnaddmediation.cnlb.kafkabroker.enable	BOOLEAN	true/ false	false	C	If true, external access for the Mediation Kafka brokers will be enabled on the CNLB enabled cluster.
ocnaddmediation.cnlb.kafkabroker.networks	STRING	NA	"default/nf-sig2-int1@nf-sig2-int1,default/nf-sig2-int2@nf-sig2-int2,default/nf-sig2-int3@nf-sig2-int3,default/nf-sig2-int4@nf-sig2-int4"	C	This property is used for populating the annotation k8s.v1.cni.cncf.io/networks Update the networks and networks_extip with the details generated by the section "Enable CNLB for Kafka Broker" from "OCNADD User Guide"

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmediation.cnlb.kafkabroker.networks_extip	STRING	NA	"nf-sig2-int1/10.10.10.10, nf-sig2-int2/10.10.10.11, nf-sig2-int3/10.10.10.12, nf-sig2-int4/10.10.10.13"	C	This property is used for populating the annotation oracle.com.cnc/cnlb Add the IP Addresses in the SAN entry as well for SSL communication
ocnaddmediation.env.admin.OCNADD_ADAPTER_UPGRADE_ENABLE	BOOLEAN	true/ false	false	M	Upgrade consumer adapter during Helm upgrade if the flag is set to true
ocnaddmediation.env.admin.OCNADD_CORR_UPGRADE_ENABLE	BOOLEAN	true/ false	false	M	Upgrade correlation service during Helm upgrade if the flag is set to true
ocnaddmediation.env.admin.OCNADD_DIAM_CORR_UPGRADE_ENABLE	BOOLEAN	true/ false	false	M	Upgrade diameter correlation service during Helm upgrade if the flag is set to true
ocnaddmediation.env.admin.OCNADD_INGRESS_ADAPTER_UPGRADE_ENABLE	BOOLEAN	true/ false	false	M	Upgrade ingress adapter during Helm upgrade if the flag is set to true
ocnaddmediation.env.admin.OCNADD_STORAGE_ADAPTER_UPGRADE_ENABLE	BOOLEAN	true/ false	false	M	Upgrade storage adapter during Helm upgrade if the flag is set to true
ocnaddmediation.kafka.kafkaReplicas	INTEGER	NA	4	M	The parameter to change the replicas for the Mediation Kafka broker.
ocnaddmediation.cluster.domainName	STRING	NA	occne-ocdd	M	Domain name of the setup where Mediation Group will be deployed
ocnaddmediation.cluster.clusterName	STRING	NA	occne-ocdd	M	Cluster name of the setup where Mediation Group will be deployed
ocnaddmediation.cluster.workergroupName	STRING	NA	wg1	M	Enter the logical name of the worker group that this Mediation group belongs to. This value must match the worker group configured for the associated Relay Agent.
ocnaddmediation.cluster.nameSpace.name	STRING	NA	ocnadd-relay	M	OCNADD Mediation Group namespace

Table 3-1 (Cont.) Global Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmediation.cluster.kafka.ocnadd_kafka_bootstrap_servers	STRING	NA	kafka-broker:9092	M	Mediation Kafka Bootstrap server for PLAINTEXT
ocnaddmediation.cluster.kafka.ocnadd_kafka_bootstrap_servers_ssl	STRING	NA	kafka-broker:9093	M	Mediation Kafka Bootstrap server for SSL
ocnaddmediation.cluster.kafka.ocnadd_kafka_bootstrap_servers_sasl	STRING	NA	kafka-broker:9094	M	Mediation Kafka Bootstrap server for SASL

3.3 Helm Hook Parameters

Table 3-2 Helm Hook Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddcopybackuppvcorigtotemphelmhook.name	STRING	-	ocnaddcopybackuppvcorigtotemphelmhook	M	Hook to copy backup from original to temporary PVC.
ocnaddcopybackuppvctemptoorighelmhook.name	STRING	-	ocnaddcopybackuppvctemptoorighelmhook	M	Hook to copy backup from temporary to original PVC
ocnaddcreatetemppvchelmhook.name	STRING	-	ocnaddcreatetemppvchelmhook	M	Hook to create temporary backup PVC during upgrade.
ocnaddhelmhook.config.auto_backup_restore_cm	STRING	-	ocnadd-configmap-auto-backup-restore	M	Name of the automatic backup restore ConfigMap
ocnaddmanagement.ocnaddhelmhook.config.name	STRING	-	helmhook-configmap	M	Name of ConfigMap
ocnaddmanagement.ocnaddhelmhook.config.rollback_name	STRING	-	helmhook-rollback-configmap	M	Name of Rollback ConfigMap
ocnaddmanagement.ocnaddhelmhook.config.upgrade_name	STRING	-	helmhook-upgrade-configmap	M	Name of Upgrade ConfigMap

Table 3-2 (Cont.) Helm Hook Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmanagement.ocnaddhelmhook.name	STRING	-	ocnaddhelmhook	M	Helm Hook Name
ocnaddpostdeleteworkergrouphelmhook.name	STRING	-	ocnaddpostdeleteworkergrouphelmhook	M	Name of the postdelete hook used during the uninstallation of the worker group
ocnaddpostdeleteworkergrouphelmhook.retry_delay	INTEGER	-	15	M	Delay between retries
ocnaddpostdeleteworkergrouphelmhook.retry_max_attempt	INTEGER	-	5	M	Maximum number of retries for invoking delete API for worker group
ocnaddpostinstallhelmhook.name	STRING	-	ocnaddpostinstallhelmhook	M	Post Install Hook Name
ocnaddpostinstallworkergrouphelmhook.name	STRING	-	ocnaddpostinstallworkergrouphelmhook	M	Name of the postinstall hook used during the installation of the worker group
ocnaddpostinstallworkergrouphelmhook.retry_delay	INTEGER	-	15	M	Delay between retries
ocnaddpostinstallworkergrouphelmhook.retry_max_attempt	INTEGER	-	5	M	Maximum number of retries for invoking the create worker group API
ocnaddpostrollbackhelmhook.name	STRING	-	ocnaddpostrollbackhelmhook	M	Post Rollback hook name
ocnaddpostupgradehelmhook.name	STRING	-	ocnaddpostupgradehelmhook	M	Post Upgrade Hook Name
ocnaddpreinstallworkergrouphelmhook.name	STRING	-	ocnaddpreinstallworkergrouphelmhook	M	Name of the preinstall hook used during the installation of the worker group
ocnaddpreinstallworkergrouphelmhook.retry_delay	INTEGER	-	15	M	Delay between retries
ocnaddpreinstallworkergrouphelmhook.retry_max_attempt	INTEGER	-	5	M	Maximum number of retries for getting the list of worker group names
ocnaddprerollbackhelmhook.name	STRING	-	ocnaddprerollbackhelmhook	M	Pre Rollback Hook Name
ocnaddpreupgradehelmhook.name	STRING	-	ocnaddpreupgradehelmhook	M	Pre Upgrade Hook Name

3.4 Aggregation Service Parameters

Table 3-3 Aggregation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
BATCH_SIZE	INTEGER	-	65536	O	The maximum amount of data to be collected before sending the batch.
CONSUMER_POLL_MS	INTEGER	-	50	O	Polling time in ms for consumer
DD_METADATA_MAP_CACHE_EXPIRY_TIME_MS	INTEGER	30ms-30s	30	O	DD metadata cache expiry timer, default is 30ms
DD_METADATA_MAP_CACHE_SCHEDULER_TIME_MS	INTEGER	5ms-2s	5	O	This timer value depends on the attribute METADATA_MAP_CACHE_EXPIRY_TIME_MS. The timer value should be adjusted up or down corresponding to increase or decrease in METADATA_MAP_CACHE_EXPIRY_TIME_MS, default is 5ms
ENABLE_AGGREGATION_COUNTER_METRICS	BOOLEAN	true/ false	true	M	Enable metrics for Aggregation service
ENQUEUE_SCP_ORIGIN_MESSAGES	BOOLEAN	true/ false	false	C	When this parameter is enabled, the SCP originated message in Model-D call model will be held by the aggregation service until received NF originated RxResponse/TxResponse messages or Timer expiry. This flag is used in case of TRANSACTION based message sequencing.
FETCH_MAX_WAIT_MS	INTEGER	-	100	O	The maximum amount of time the server will block before answering the fetch request if there isn't sufficient data to immediately satisfy the requirement given by fetch.min.bytes
HEARTBEAT_INTERVAL_MS	INTEGER	-	5000	O	The expected time between heartbeats to the consumer coordinator when using Kafka's group management facilities
KAFKA_COMMIT_INT_CONFIG	INTEGER	-	1000	O	The frequency at which consumer offsets are committed to the Kafka broker
KAFKA_FETCH_MAX_BYTES	STRING	-	57672000	O	The maximum amount of data per-partition the server will return
KAFKA_FETCH_MIN_BYTES	STRING	-	1	O	The minimum amount of data per-partition the server will return
KAFKA_MAX_AGE_CONFIG	INTEGER	-	7500	M	The period of time in milliseconds after which we force a refresh of metadata.

Table 3-3 (Cont.) Aggregation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
KAFKA_MAX_PARTITIONS_FETCH_BYTES	STRING	-	10485800	O	The maximum amount of data per-partition the server will return.
KAFKA_PRODUCER_SSL_CLIENT_AUTH	BOOLEAN	true/ false	false	M	Kafka SSL client authentication.
KAFKA_SOCKET_BYTES_BUFFER	STRING	-	10485700	O	Kafka Socket Buffer setting for consumer
LINGER_MS	INTEGER	-	1	O	The time to wait before sending messages out to Kafka
MAX_POLL_INTERVAL_MS	INTEGER	-	30000	O	The maximum delay between invocations of poll() when using consumer group management
MAX_POLL_RECORDS	INTEGER	-	300	O	The maximum number of records returned in a single call to poll()
MESSAGE_REORDERING_INCOMPLETE_TRANSACTION_METRICS_ENABLE	BOOLEAN	true/ false	false	O	The parameter enables the Metric to check/count missing/inordered messages of transactions for MESSAGE_SEQUENCING_TYPE=TRANSACTION/REQUEST_RESPONSE
MESSAGE_SEQUENCING_TYPE	INTEGER	[NONE, TIME_WINDOW, TRANSACTION ,REQUEST_RESPONSE]	NONE	M	NONE: No message sequencing. TIME_WINDOW: Messages received within window time for each partition will be sorted separately based on time stamp and stream to kafka topic. TRANSACTION: In order messages received for each transaction within TRANSACTION_EXPIRY_TIME will be sorted separately and stream to kafka topic. REQUEST_RESPONSE: In order Request(RxRequest and TxRequest) and/or Response pair(RxResponse and TxResponse) messages received for each transaction within REQUEST_RESPONSE will be sorted separately and stream to kafka topic.
OCNADD_AGGREGATION_REDUNDANCY_BUFFER_ENABLED	BOOLEAN	true/ false	false	O	When enabled, this parameter allows the system to buffer and retry messages that fail to send to the Mediation Kafka broker, ensuring they are successfully forwarded. Note that enabling this parameter requires sufficient memory to be allocated to the aggregation service.
OCNADD_AGGREGATION_SERVICE_TOPIC_RETRIES_THRESHOLD	INTEGER	-	120000	O	Retry Threshold for TOPIC reachability

Table 3-3 (Cont.) Aggregation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
OCNADD_ASR_CLIENT_ENABLE_NETWORK_LATENCY_METRICS	BOOLEAN	true/ false	false	O	The parameter enables network latency metrics for aggregation service.
OCNADD_KAFKA_SECURITY_PROTOCOL_SSL	STRING		SSL	M	Kafka SSL Mechanism.
ocnaddbsfaggregation.maxReplicas	INTEGER	-	1	M	The maximum number of replicas required for BSF aggregation service instance
ocnaddbsfaggregation.minReplicas	INTEGER	-	1	M	The minimum number of replicas required for BSF aggregation service instance
ocnaddbsfaggregation.name	STRING	-	ocnaddbsfaggregation	M	Name of the application
ocnaddbsfaggregation.resources.limit.cpu	INTEGER	-	2	M	Number of max CPU for BSF Aggregation
ocnaddbsfaggregation.resources.limit.ephemeralstorage	STRING	-	500Mi	M	Ephemeral Storage for BSF Aggregation
ocnaddbsfaggregation.resources.limit.memory	STRING	-	2Gi	M	Max Memory limit for BSF Aggregation
ocnaddnrfaggregation.maxReplicas	INTEGER	-	1	M	The maximum number of replicas required for NRF aggregation service instance
ocnaddnrfaggregation.minReplicas	INTEGER	-	1	M	The minimum number of replicas required for NRF aggregation service instance
ocnaddnrfaggregation.name	STRING	-	ocnaddnrfaggregation	M	Name of the application
ocnaddnrfaggregation.resources.limit.cpu	INTEGER	-	3	M	Number of maximum CPUs for NRF aggregation
ocnaddnrfaggregation.resources.limit.ephemeralstorage	STRING	-	500Mi	M	Ephemeral Storage for NRF Aggregation
ocnaddnrfaggregation.resources.limit.memory	STRING	-	2Gi	M	Max Memory limit for NRF Aggregation
ocnaddpcfaggregation.maxReplicas	INTEGER	-	2	M	The maximum number of replicas required for PCF aggregation service instance

Table 3-3 (Cont.) Aggregation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddpcfaggregation.minReplicas	INTEGER	-	1	M	The minimum number of replicas required for PCF aggregation service instance
ocnaddpcfaggregation.name	STRING	-	ocnaddpcfaggregation	M	Name of the application
ocnaddpcfaggregation.resources.limit.cpu	INTEGER	-	2	M	Number of max CPU for PCF Aggregation
ocnaddpcfaggregation.resources.limit.ephemeralstorage	STRING	-	500Mi	M	Ephemeral Storage for PCF Aggregation
ocnaddpcfaggregation.resources.limit.memory	STRING	-	2Gi	M	Max Memory limit for PCF Aggregation
ocnaddscpaggregation.maxReplicas	INTEGER	-	4	M	The maximum number of replicas required for SCP aggregation service instance
ocnaddscpaggregation.minReplicas	INTEGER	-	1	M	The minimum number of replicas required for SCP aggregation service instance
ocnaddscpaggregation.name	STRING	-	ocnaddscpaggregation	M	Name of the application
ocnaddscpaggregation.resources.limit.cpu	INTEGER	-	3	M	Number of max CPU for SCP Aggregation
ocnaddscpaggregation.resources.limit.ephemeralstorage	STRING	-	500Mi	M	Ephemeral Storage for SCP Aggregation
ocnaddscpaggregation.resources.limit.memory	STRING	-	2Gi	M	Max Memory limit for SCP Aggregation
ocnaddseppaggregation.maxReplicas	INTEGER	-	2	M	The maximum number of replicas required for SEPP aggregation service instance
ocnaddseppaggregation.minReplicas	INTEGER	-	1	M	The minimum number of replicas required for SEPP aggregation service instance
ocnaddseppaggregation.name	STRING	-	ocnaddseppaggregation	M	Name of the application
ocnaddseppaggregation.resources.limit.cpu	INTEGER	-	3	M	Number of max CPU for SEPP Aggregation

Table 3-3 (Cont.) Aggregation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddseppaggregation.resources.limit.ephemeralstorage	STRING	-	500Mi	M	Ephemeral Storage for SEPP Aggregation
ocnaddseppaggregation.resources.limit.memory	STRING	-	2Gi	M	Max Memory limit for SEPP Aggregation
ocnaddvcollectordiameteraggregation.maxReplicas	INTEGER	-	1	M	The maximum number of replicas required for ocnaddvcollectordiameteraggregation aggregation service instance
ocnaddvcollectordiameteraggregation.minReplicas	INTEGER	-	1	M	The minimum number of replicas required for ocnaddvcollectordiameteraggregation aggregation service instance
ocnaddvcollectordiameteraggregation.name	STRING	-	ocnaddvcollectordiameteraggregation	M	Name of the application
ocnaddvcollectordiameteraggregation.resources.limit.cpu	INTEGER		2	M	Number of max CPU for vCollector Diameter Aggregation
ocnaddvcollectordiameteraggregation.resources.limit.ephemeralstorage	STRING		500Mi	M	Ephemeral Storage for vCollector Diameter Aggregation
ocnaddvcollectordiameteraggregation.resources.limit.memory	STRING		2Gi	M	Max Memory limit for vCollector Diameter Aggregation
PRODUCERS_BUFFER_MEMORY	INTEGER	-	67108864	O	The total bytes of memory the producer can use to buffer records waiting to be sent to the server.
PRODUCERS_COMPRESSION_TYPE	STRING	none/ snappy	snappy	O	The compression type for all data generated by the producer. Changing compression type to none may impact latency and throughput when running OCNADD under high traffic load.
PRODUCERS_DELIVERY_TIMEOUT_MS	INTEGER	-	120000	O	The maximum amount of time, in milliseconds, that a Kafka producer will wait for a message to be successfully delivered to the broker and acknowledged.
PRODUCERS_MAX_BLOCK_MS	INTEGER	-	10000	O	The maximum amount of time, in milliseconds, that the Kafka producer will block when attempting to send a record.

Table 3-3 (Cont.) Aggregation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
PRODUCERS_MAX_REQUEST_SIZE	INTEGER	-	1048500	O	The maximum size of a request in bytes. This setting will limit the number of record batches the producer will send in a single request to avoid sending huge requests.
PRODUCERS_RETRIES	INTEGER	-	0	O	The number of times producer will retry a request that may fail due to transient error. Configuring this property to more than 0 may impact in throughput
REQUEST_RESPONSE_MSG_SEQUENCING_EXPIRY_TIMER	INTEGER	[5-500]ms	10ms	C	When MESSAGE_SEQUENCING_TYPE = REQUEST_RESPONSE Range [5ms-500ms]; default: 10ms
REQUEST_TIMEOUT_MS	INTEGER	-	1000	O	The configuration controls the maximum amount of time the client will wait for the response of a request
SESSION_TIME_OUT	INTEGER	-	15000	O	The timeout used to detect client failures when using Kafka's group management facility.
SRC_FEED_METADATA_CLEAN_DELAY_SEC	INTEGER	-	86400	C	Initial delay in cleaning the metadata cache
SRC_FEED_METADATA_CLEAN_PERIOD_SEC	INTEGER	-	86400	C	Interval after the metadata cache is cleaned
SRC_FEED_METADATA_HISTORY_LENGTH	INTEGER	-	20	C	The number of the metadata maintained in the cache
TOTAL_FORWARDED_MESSAGE_METRICS_ENABLE	BOOLEAN	true/ false	false	O	The parameter enables the Metric to get total count of forwarded messages by aggregation service
TRANSACTION_MSG_SEQUENCING_EXPIRY_TIMER	INTEGER	[20ms-30s]	200ms	C	When MESSAGE_SEQUENCING_TYPE = TRANSACTION Range [20ms-30s]; default: 200ms
WINDOW_MSG_SEQUENCING_EXPIRY_TIMER	INTEGER	[5-500]ms	10ms	C	When MESSAGE_SEQUENCING_TYPE = TIME_WINDOW Range [5ms-500ms]; default: 10ms

3.5 Configuration Service Parameters

Table 3-4 Configuration Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddconfiguration.name	STRING	-	ocnaddconfiguration	M	Name of configuration service
MAX_ALLOWED_FILTERS	INTEGER	-	30	O	Maximum number of allowed filters
MAX_VALUES_IN_FILTER_ATTRIB	INTEGER	-	20	O	Maximum number of values allowed in filter attributes
MAX_FILTERS_ASSOC_WITH_APP	INTEGER	-	4	O	Maximum number of filters associated with a single app
MAX_ACTION_TYPE_IN_FILTER	INTEGER	-	2	O	Maximum number of action type in a filter with chaining
MAX_EXTERNAL_AGGREGATED_FEEDS	INTEGER	-	2	O	Maximum number of allowed external aggregated feeds
MAX_L3L4_ATTRIBUTES	INTEGER	-	2	O	Maximum number of allowed L3L4 attributes
MAX_GLOBALL3L4_ROW_SIZE	INTEGER	-	500	O	Maximum size of L3L4 rows
MAX_CORRELATION_CONFIGURATION_SUPPORTED	INTEGER	-	3	O	Maximum number of correlation feeds allowed
OCNADD_MAX_WORKERGROUP_THRESHOLD_PERCENTAGE	INTEGER	-	80	O	The percentage threshold for the maximum worker group supported
MAX_EXPORT_CONFIGURATION_SUPPORTED	INTEGER	-	3	O	The simultaneous number of export configuration supported on the Data Director
EXPORT_CONFIG_PURGE_TIMER_HR	INTEGER	[1-48]	24	O	The purge timer for the export configuration
EXPORT_PURGE_SCHEDULER_DELAY_MS	INTEGER	-	30000	O	The delay in milisec after which the purging of the export configuration will be checked.
CONFIG_NOTIFICATION_SCHEDULER_PERIOD_MS	STRING	-	200ms	O	The notification sender thread fixed delay upon which it will check for the pending notiifications

Table 3-4 (Cont.) Configuration Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
CONFIGURATION_FILTER_METHODS	STRING	[GET,POST,PUT,DELETE,PATCH,CONNECT,OPTIONS,TRACE]	GET,POST,PUT,DELETE,PATCH,CONNECT,OPTIONS,TRACE	O	The configuration of methods on which the filter is possible
ADAPTER_DEFAULT_INGRESS_TOPIC	STRING	-	MAIN	O	The topic name for the non oracle ingress feed adapter
ADAPTER_DEFAULT_INGRESS_PARTITIONS	INTEGER	-	30	O	The number of partitions for the non oracle ingress feed adapter
ADAPTER_DEFAULT_INGRESS_REPLICATION_FACTOR	INTEGER	-	1	O	The non oracle topic replication factor
ADAPTER_DEFAULT_INGRESS_RETENTION_MS	INTEGER	-	600000	O	The non oracle topic retention time in milisec
ADAPTER_DEFAULT_INGRESS_ACKS	STRING	-	all	O	The non oracle producer acknowledgement value
ADAPTER_DEFAULT_INGRESS_RETRY	INTEGER	-	3	O	The number of retries for the non oracle ingress adapter feed
ADAPTER_DEFAULT_INGRESS_LIMIT	INTEGER	-	101874	O	The buffer in bytes in the non oracle ingress adapter feed for the http connection
resources.limits.cpu	INTEGER	-	1	M	Number of maximum CPUs for each configuration service instance
resources.limits.memory	STRING	-	1Gi	M	Max Memory limit for each configuration service instance
resources.limits.ephemeralstorage	STRING	-	500Mi	M	Ephemeral Storage for each configuration service
resources.requests.cpu	INTEGER	-	1	M	Minimum number of CPUs required for each configuration service instance
resources.requests.memory	STRING	-	1Gi	M	minimum Memory required for each configuration instance
resources.requests.ephemeralstorage	STRING	-	100Mi	M	minimum Ephemeral Storage required for each configuration instance
ocnaddmigration.name	STRING	-	ocnaddmigration	M	Name of the migration Job
ocnaddmigration.resources.limit.cpu	STRING	-	500m	M	Maximum CPU required for the Migration Job

Table 3-4 (Cont.) Configuration Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmigration.resources.limit.memory	STRING	-	1Gi	M	Maximum Memory required for the Migration Job
ocnaddmigration.resources.requests.cpu	STRING	-	500m	M	Minimum CPU required for the Migration Job
ocnaddmigration.resources.requests.cpu	STRING	-	1Gi	M	Minimum Memory required for the Migration Job

3.6 Health Monitoring and Alarm Service Parameters

Table 3-5 Health Monitoring Service Parameters

Parameter Name	Data Type	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddhealthmonitoring.name	STRING	ocnaddhealthmonitoring	M	Health monitoring service name
HEALTH_MONITORING_TIMER	INTEGER	5000	O	Timer to check Health of integrated services
HEALTH_METRICS_SCHEDULED	BOOLEAN	true	O	Scheduler for metrics
HEALTH_METRICS_TIMER	INTEGER	120000	O	Timer for health metrics
HEALTH_PURGE_TIME_HR	INTEGER	5	O	Health profile purging timer in hour
HEALTH_MONITORING_CPUTHRESHOLD	INTEGER	75	M	CPU threshold to raise alarm
HEALTH_MONITORING_MEMORYTHRESHOLD	INTEGER	95	M	Memory threshold to raise alarm
Logging Properties				
resources.limits.cpu	INTEGER	1	M	Number of maximum CPUs for each health monitoring service instance
resources.limits.memory	STRING	1Gi	M	Max Memory limit for each health monitoring service instance
resources.limits.ephemeralstorage	STRING	500Mi	M	Ephemeral Storage for each health monitoring service
resources.requests.cpu	INTEGER	1	M	Minimum number of CPUs required for each health monitoring service instance

Table 3-5 (Cont.) Health Monitoring Service Parameters

Parameter Name	Data Type	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
resources.requests.memory	STRING	1Gi	M	minimum Memory required for each health monitoring instance
resources.requests.ephemeralstorage	STRING	200Mi	M	minimum Ephemeral Storage required for each health monitoring instance

Table 3-6 Alarm Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddalarm.name	STRING	-	ocnaddalarm	M	Application name for Alarm Service
resources.limits.cpu	INTEGER	-	1	M	Number of maximum CPUs for each alarm service instance
resources.limits.memory	STRING	-	1Gi	M	Max Memory limit for each alam service instance
resources.limits.ephemeralstorage	STRING	-	200Mi	M	Ephemeral Storage for each alarm service
resources.requests.cpu	INTEGER	-	1	M	Minimum number of CPUs required for each alarm service instance
resources.requests.memory	STRING	-	1Gi	M	minimum Memory required for each alarm instance
resources.requests.ephemeralstorage	STRING	-	200Mi	M	minimum Ephemeral Storage required for each alarm instance

Table 3-6 (Cont.) Alarm Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
MAX_ALARM_RETRIEVE_COUNT	INTEGER	-	1000	O	Parameter to fetch maximum of 1000 latest alarm for each severity type

3.7 Admin Service Parameters

Table 3-7 Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
resources.limits.cpu	INTEGER	-	1	M	Number of maximum CPUs for each admin service
resources.limits.memory	STRING	-	1Gi	M	Max Memory limit for each admin service instance
resources.limits.ephemeralstorage	STRING	-	200Mi	M	Ephemeral Storage for each admin service
resources.requests.cpu	INTEGER	-	1	M	Minimum number of CPUs required for each admin service instance
resources.requests.memory	STRING	-	1Gi	M	minimum Memory required for each admin instance
resources.requests.ephemeralstorage	STRING	-	200Mi	M	minimum Ephemeral Storage required for each admin instance
OCNADD_EGRESS_NETWORK_ENABLE	BOOLEAN	true/ false	false	O	Enable this parameter to true if traffic needs to be routed outside the cluster.
OCNADD_EGRESS_NETWORK_NAME_VALUE	STRING	-	oam	O	Name of the egress network configured in the CNE cluster.
OCNADD_CNC_ENABLE	STRING		True	O	Enable oracle.com.cnc network.
Consumer Aapter Parameters					
consumeradapter.resources.limits.cpu	INTEGER	-	3	M	Number of maximum CPUs for each admin service
consumeradapter.resources.limits.memory	STRING	-	6Gi	M	Max Memory limit for each admin service instance
consumeradapter.resources.limits.ephemeralstorage	STRING	-	1Gi	M	Ephemeral Storage for each admin service
consumeradapter.resources.requests.cpu	INTEGER	-	3	M	Minimum number of CPUs required for each admin service instance
consumeradapter.resources.requests.memory	STRING	-	6Gi	M	minimum Memory required for each Correlation instance

Table 3-7 (Cont.) Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
consumeradapter.resources.requests.ephemeralstorage	STRING	-	1Gi	M	minimum Ephemeral Storage required for each Correlation instance
EGRESS_SSL_HANDSHAKE_TIMEOUT	INTEGER	-	5	O	SSL handshake timeout.
ENABLE_L3L4_COUNTER_METRICS	BOOLEAN	true/ false	false	O	Enable L3L4 Metric.
OCNADD_ADAPTER_MIN_REPLICAS_HTTP2	INTEGER	-	2	M	Minimum Replicas for HTTP2 Adapter
OCNADD_ADAPTER_MAX_REPLICAS_HTTP2	INTEGER	-	13	M	Max Replicas for HTTP2 Adapter
OCNADD_ADAPTER_MIN_REPLICAS_TCP	INTEGER	-	1	M	Minimum Replicas for TCP Adapter
OCNADD_ADAPTER_MAX_REPLICAS_TCP	INTEGER	-	9	M	Max Replicas for TCP Adapter
MAX_TCP_CONNECTION_PER_DEST	INTEGER	-	6	M	Max allowed TCP connection per destination
ADAPTER_KAFKA_FETCH_MIN_BYTES	INTEGER	-	0	O	The minimum amount of data the server should return for a fetch request
ADAPTER_KAFKA_FETCH_MAX_BYTES	STRING	-	57672000	O	The maximum amount of data the server should return for a fetch request
ADAPTER_KAFKA_MAX_PARTITION_FETCH_BYTES	STRING	-	10485700	O	The maximum amount of data per-partition the server will return
ADAPTER_KAFKA_FETCH_MAX_WAIT_MS	INTEGER	-	40	O	The maximum amount of time the server will block before answering the fetch request
ADAPTER_KAFKA_SESSION_TIME_OUT	INTEGER	-	15000	O	The timeout used to detect client failures when using Kafka's group management facility
ADAPTER_KAFKA_HEARTBEAT_INTERVAL_MS	INTEGER	-	5000	O	The expected time between heartbeats to the consumer coordinator when using Kafka's group management facilities
ADAPTER_KAFKA_MAX_POLL_INTERVAL_MS	INTEGER	-	30000	O	The maximum delay between invocations of poll() when using consumer group management
ADAPTER_KAFKA_MAX_POLL_RECORDS	INTEGER	-	1500	O	The maximum number of records returned in a single call to poll()

Table 3-7 (Cont.) Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
ADAPTER_KAFKA_COMMIT_INT_CONFIG	INTEGER	-	30	O	The frequency in milliseconds that the consumer offsets are committed to Kafka
ADAPTER_KAFKA_NUM_THREADS_CONFIG_HTTP2	INTEGER	-	9	O	The number of threads to run stream processing for http2 connections.
ADAPTER_KAFKA_NUM_THREADS_CONFIG_TCP	INTEGER	-	6	O	The number of threads to run stream processing for tcp connections.
ADAPTER_KAFKA_CONSUMER_POLL_MS	INTEGER	-	30	O	The amount of time in milliseconds to block waiting for input.
ADAPTER_KAFKA_RECEIVE_BUFFER_BYTES	STRING	-	10485700	M	The size of the TCP receive buffer (SO_RCVBUF) to use when reading data.
ADAPTER_LOG_LEVEL_KAFKA	STRING	[ON,OFF]	OFF	O	Whether to ON or OFF Kafka logs in Adapter Service.
OCNADD_ADAPTER_MAX_REPLICAS_TCP	INTEGER		2	O	MAX replicas for synthetic Adapter.
OCNADD_ADAPTER_LIVENESS_DELAY	INTEGER		60	M	Adapter Svc Liveness Param: this field tells the kubelet that it should wait for mentioned seconds before performing the first probe.
OCNADD_ADAPTER_LIVENESS_PERIOD_SECONDS	INTEGER		15	M	Adapter Svc Liveness Param: this field specifies that the kubelet should perform a liveness probe every given number of seconds.
OCNADD_ADAPTER_LIVENESS_FAILURE	INTEGER		5	M	Adapter Svc Liveness Param: For the case of a liveness probe, triggers a restart for that specific container if the container failed to start for given number of failure retries.
OCNADD_ADAPTER_LIVENESS_TIMEOUT	INTEGER		20	M	Adapter Svc Liveness Param: Number of seconds after which the probe times out.
CONFIG_SVC_DATASTREAM_OFFSET_DELAY_MS	INTEGER		3000	O	Delay in milliseconds between Retries to fetch the data stream offset from config service in case of failure.
ADAPTER_KAFKA_LISTCONSUMER_TIMEOUT_MS	INTEGER		30000	O	Timeout in milliseconds to list the Consumer Groups.
KAFKA_TOPIC_NO_OF_PARTITIONS	INTEGER		42	O	Default number of partitions that will be created for a topic when a new Kafka feed is created.
KAFKA_TOPIC_REPLICATION_FACTOR	INTEGER		1	O	Replication Factor for the Kafka Topic of Kafka Feed.
KAFKA_TOPIC_RETENTION_MS	INTEGER		300000	O	Retention Time for Kafka Topic.

Table 3-7 (Cont.) Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
KAFKA_TOPIC_SEGMENT_MS	INTEGER		300000	O	Retention Time for the Kafka topic segment
EXTERNAL_CONSUMER_SASL_PORT	INTEGER		9094	O	Kafka bootstrap server port number for SASL_SSL protocol.
KAFKA_FUTURE_TIMEOUT_MS	INTEGER		25000	O	Timeout to fetch the admin client response details.
VERIFY_KAFKA_CONNECTION_TIMEOUT_MS	INTEGER		10000	O	Timeout to verify Kafka connection in milliseconds.
KAFKAFEED_METRICS_SCHEDULED	BOOLEAN		true	O	To enable or disable metrics for Kafka Feeds.
KAFKAFEED_METRICS_TIMER	STRING		15s	O	Metrics timer for Kafka Feeds.
TCP_STREAM_RESET_ENABLED	BOOLEAN		false	O	To enable Kafka feed stream restart. This maybe required only when 3rd Party consumer is not working properly and frequently breaks connections with Synthetic Feed
TCP_STREAM_RESET_ELAPSED_TIME	INTEGER	-	60	O	The time in minutes to check if stream reset is required.
TCP_STREAM_RESET_FIXED_DELAY_MS	INTEGER		300000	O	Default 300 sec, Schedular Interval Time
TCP_STREAM_RESET_INI_DELAY_MS	INTEGER		150000	O	Default 150 sec, Schedular Initial Delay to Start
TCP_STREAM_RESTART_INTERVAL_MS	INTEGER		300000	O	Default 300 sec, interval to check for the TCP stream restart
TCP_CONN_INSPECTOR_ENABLED	BOOLEAN	true/ false	true	O	The parameter to enable the watcher for the TCP connection in consumer adapter
OCNADD_INTERNAL_CLIENT_SSL_PROTOCOL	STRING		TLS	O	The SSL protocol used between the adapter and internal DD services communication
OCNADD_INTERNAL_CLIENT_SSL_PROTOCOLS	STRING		TLSv1.2, TLSv1.3	O	The version of TLS supported between adapter and internal DD services communication
OCNADD_INTERNAL_CLIENT_SSL_HANDSHAKE_TIMEOUT	STRING		30s	O	SSL handshake timeout between adapter and internal DD services communication

Table 3-7 (Cont.) Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
OCNADD_SSL_CIPHERS	STRING		TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	O	Oracle Ciphers supported for the TLS communication
OCNADD_ADAPTER_CONFIG_RETRY_DELAY	STRING		5s	M	The retry interval between adapter and configuration service communication for subscription
OCNADD_ADAPTER_CONFIG_RETRY_DELETE_SUBS_REQ_ATTEMPT	STRING		3	M	The number of retries supported for sending the delete subscription request to configuration service by consumer adapter service
OCNADD_ADAPTER_CONFIG_RETRY_DELETE_SUBS_REQ_DELAY	STRING		3s	M	The retry interval supported for sending the delete subscription request to configuration service by consumer adapter service
EGRESS_TRUSTSTORE_ENABLED	BOOLEAN	true/false	true	M	The parameter to enable/disable the truststore between consumer adapter and 3rd party application
EGRESS_SSL_CLIENT_AUTH	STRING	need/want	want	M	SSL Authentication mode to be supported between consumer adapter and 3rd party communication
EGRESS_SSL_HANDSHAKE_TIMEOUT	STRING	-	5s	M	SSL handshake timeout between adapter and 3rd party application communication
HTTP_ENABLE_SUBSCRIBE_API_ASYNC_PROCESSING	BOOLEAN	true/false	true	M	The parameter to enable the asynchronous processing of subscription request towards the configuration service
EGRESS_RESPONSE_IDLE_TIMEOUT_SEC	INTEGER	-	120	O	The setting (which defaults to 2 minutes) dictates when to close a connection after it becomes idle
EGRESS_RESPONSE_TIMEOUT_SEC	INTEGER	-	8	O	The amount of time it takes to actually receive the response back from the server, default is 8 sec

Table 3-7 (Cont.) Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
EGRESS_CHANNEL_TIMEOUT_SEC	INTEGER	-	6	O	Specifies the amount of time, in seconds, that the HTTP transport channel waits for a read request to complete on a socket after the first read occurs. Default is 6 sec
EGRESS_HTTP_FOLLOW_REDIRECTS	STRING	-	false	O	The parameter to indicate that consumer adapter does not want any redirections from the 3rd party applications
SO_SEND_BUF_BYTES_HTTP	STRING		16777216	O	The send socket buffer size for the HTTP connection towards 3rd party application
EGRESS_KEEPALIVE_IDLE	INTEGER	-	60	O	The keepalive will be sent after the connection is idle for the configured seconds
EGRESS_KEEPALIVE_INT	INTEGER	-	60	O	interval between two keepalive messages
EGRESS_KEEPALIVE_COUNT	INTEGER	-	10	O	The maximum number of keepalive packets that will be sent before assuming the connection is dead
EGRESS_HTTP_CLIENT_MAX_CONCURRENT_HTTP_CONN	INTEGER	-	100	O	The maximum number of concurrent HTTP connection that can be made by http client
EGRESS_HTTP_MAX_CONCURR_REQ_PER_HTTP_CONN	INTEGER	-	5	O	The maximum number of concurrent HTTP Requests that can be sent by http client on the single http connection
EGRESS_MAX_CONNECTION_POOL_IDLE	INTEGER	-	30	O	In the http client connection pool, the connections that are not currently in use but are maintained by the pool for reuse
EGRESS_HTTP_CLIENT_SHUTDOWN_QUIET_PERIOD_SEC	INTEGER	-	25	O	Sets the amount of quiet period for shutdown of client thread pools
EGRESS_HTTP_CLIENT_SHUTDOWN_TIMEOUT_SEC	INTEGER	-	30	O	Sets the amount of time to wait for shutdown of client thread pools
ADAPTER_TCP_CLIENT_POOL_MAX	INTEGER	-	1000	O	the maximum number of concurrent TCP connections a client can establish with a server
ADAPTER_TCP_CLIENT_CHANNEL_TIMEOUT	INTEGER	-	60	O	A TCP client channel timeout occurs when a TCP client doesn't receive a response from a server within a specific timeframe, leading to the connection being terminated
ADAPTER_TCP_CLIENT_KEEPALIVE_IDLE	INTEGER	-	120	O	The keepalive will be sent after the connection is idle for the configured seconds
ADAPTER_TCP_CLIENT_KEEPALIVE_INT	INTEGER	-	20	O	interval between two keepalive messages

Table 3-7 (Cont.) Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
ADAPTER_TCP_CLIENT_KEEPALIVE_COUNT	INTEGER	-	5	O	The maximum number of keepalive packets that will be sent before assuming the connection is dead
ADAPTER_TCP_CLIENT_SSL_HANDSHAKE	INTEGER	-	60	O	The timeout for the SSL handshake to complete for the TCP connection
TCP_CONN_INSPECTOR_FIXED_DELAY	STRING	-	30s	O	The delay interval after which the TCP connection watcher will keep on watching the connection
TCP_CONN_INSPECTOR_INI_DELAY	STRING	-	20s	O	The initial delay after which the TCP connection watcher will start watching the connection
ADAPTER_TCP_CONNECT_TIMEOUT	INTEGER	-	60	O	The TCP connection timeout after which there is no response received from the server
ADAPTER_TCP_CONNECT_RETRY_MS	INTEGER	-	1000	O	The TCP connection retry interval
ADAPTER_TCP_CONNECT_MAX_RETRY_DELAY_MS	STRING	-	120000	O	The maximum wait for the TCP connection retry
TCP_SEQ_ACK_CONNECTION_MGMT_TIMER_MILLI	STRING	-	7200000	O	The connection management timer for the sequence acknowledgement feature in consumer adapter. It is configured in milisec
SYNTHETIC_SEQ_ACK_CACHE_CLEAN_DELAY_SEC	INTEGER	-	5	O	The timeout after which the cache cleaning will happen for the sequence ack cache
SYNTHETIC_SEQ_ACK_CACHE_CLEAN_PERIOD_SEC	INTEGER		5	O	The time period an entry will remain in sequence ack cache
SYNTHETIC_STREAM_ID_TRANSACTION_MGMT_TIMER_MILLI	INTEGER		100000	O	The connection management timer for the stream-id feature in consumer adapter. It is configured in milisec
SYNTHETIC_STREAM_ID_CACHE_CLEAN_DELAY_SEC	INTEGER		2	O	The timeout after which the cache cleaning will happen for the stream-id cache
SYNTHETIC_STREAM_ID_CACHE_CLEAN_PERIOD_SEC	INTEGER		2	O	The time period an entry will remain in stream-id cache
ADAPTER_KAFKA_SOURCE_TOPIC	STRING	-	MAIN	M	The topic name that consumer adapter will start reading from
ADAPTER_KAFKA_TOPIC_CHECK_INITIAL_DELAY	STRING	-	120s	M	The interval after which the consumer adapter will check if the topic to consume has been created.
OCNADD_KAFKA_SSL_PROTOCOL	STRING	-	TLSv1.3	M	SSL protocol configured in consumer adapter with Kafka cluster

Table 3-7 (Cont.) Admin Service Parameters

Parameter Name	Data Type	Range	Default Value	M/O/C	Description
ADAPTER_KAFKA_MAX_METADATA_AGE	STRING	-	300000	O	The time after which the consumer adapter will refresh the metadata information from the Kafka cluster
ENABLE_KAFKA_RECORD_TIMESTAMP_PROCESSING	BOOLEAN	true/ false	true	O	The parameter to denote that Kafka record timestamp should be used in the latency calculation
ADAPTER_KAFKA_ENABLE_AUTO_COMMIT	BOOLEAN	true/ false	false	O	The parameter to enable the autocommit of Kafka consumer offsets by the consumer adapter
ADAPTER_KAFKA_AUTOCOMMIT_INT_CONFIG	INTEGER	-	0	O	The value of autocommit interval if autocommit is enabled in the consumer adapter
STREAM_THREAD_DELAY_MS	INTEGER	-	10000	O	The initial delay in the stream processing after which the asynchronous retries will be tried.
ADAPTER_ASYNC_ENDPOINT_RETRY_SCH_FXD_DELAY	STRING	-	30s	O	The delay after which the retry is done in an asynchronous communication with the 3rd party application
MAX_CONSECUTIVE_ERRORS_ALLOWED	INTEGER	-	10	O	The number of consecutive failure after which the circuit is broken
MAX_CONSECUTIVE_SUCCESS_REQ_ALLOWED	STRING	-	270000	O	The consecutive successful messages count before the circuit is deemed to be closed again

3.7.1 Correlation Service Parameters

All the Correlation parameters are available under `ocnaddadmin.correlation` section in `ocnadd-mediation-custom-values.yam` file.

Table 3-8 Correlation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
correlation.resources.limits.cpu	INTEGER	-	3	M	Number of maximum CPUs for each Correlation instance
correlation.resources.limits.memory	STRING	-	64Gi	M	Max Memory limit for each Correlation instance

Table 3-8 (Cont.) Correlation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
correlation.resources.limits.ephemeralstorage	STRING	-	800Mi	M	Ephemeral Storage for each Correlation instance
correlation.resources.requests.cpu	INTEGER	-	3	M	Minimum number of CPUs required for each Correlation instance
correlation.resources.requests.memory	STRING	-	24Gi	M	minimum Memory required for each Correlation instance
correlation.resources.requests.ephemeralstorage	STRING	-	400Mi	M	minimum Ephemeral Storage required for each Correlation instance
correlation.minReplicas	INTEGER	-	1	M	Minimum replicas of correlation service instance
correlation.maxReplicas	INTEGER	-	4	M	Maximum replicas of correlation service instance
Environmental variables for Correlation service are declared under "ocnaddadminsvc.correlation.env" section					
ADMIN_CORRELATION_RESOURCE_FILE	STRING	-	/tmp/ocnadd/deploy/ocnaddcorrelationservice.yaml	M	Template file for deploying correlation service through admin service.
KAFKA_STREAM_STATE	STRING	-	/tmp/ocnadd/kafka/state	O	temporary storage for kafka state store
KAFKA_REPLICATION_FACTOR	INTEGER	-	1	O	replication factor for state store
KAFKA_ENABLE_AUTO_COMMIT	BOOLEAN	[true/false]	false	O	enable or disable kafka auto commit
KAFKA_SOCKET_BYTES_BUFFER	INTEGER	-	104857	O	Kafka Socket Buffer setting for consumer
KAFKA_SOCKET_BYTES_BUFFER_PORTION	INTEGER	-	100	O	This parameter is used to multiply with KAFKA_SOCKET_BYTES_BUFFER

Table 3-8 (Cont.) Correlation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
KAFKA_FETCH_MIN_BYTES	INTEGER	-	1	O	The minimum amount of data per-partition the server will return
KAFKA_FETCH_MAX_BYTES	INTEGER	-	576720	O	The maximum amount of data per-partition the server will return
KAFKA_FETCH_MAX_BYTES_PORTION	INTEGER	-	100	O	This parameter is used to multiply with KAFKA_FETCH_MAX_BYTES
KAFKA_MAX_PARTITIONS_FETCH_BYTES	INTEGER	-	104858	O	The maximum amount of data per-partition the server will return.
KAFKA_MAX_PARTITIONS_FETCH_BYTES_PORTION	INTEGER	-	10	O	This parameter is used to multiply with KAFKA_MAX_PARTITIONS_FETCH_BYTES
FETCH_MAX_WAIT_MS	INTEGER	-	100	O	The maximum amount of time the server will block before answering the fetch request if there isn't sufficient data to immediately satisfy the requirement given by fetch.min.bytes
SESSION_TIME_OUT	INTEGER	-	15000	O	The timeout used to detect client failures when using Kafka's group management facility.
HEARTBEAT_INTERVAL_MS	INTEGER	-	5000	O	The expected time between heartbeats to the consumer coordinator when using Kafka's group management facilities

Table 3-8 (Cont.) Correlation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
MAX_POLL_INTERVAL_MS	INTEGER	-	30000	O	The maximum delay between invocations of poll() when using consumer group management
MAX_POLL_RECORDS	INTEGER	-	500	O	The maximum number of records returned in a single call to poll()
KAFKA_OFFSET_CONFIG	STRING	-	latest	O	default kafka data stream offset config
KAFKA_AUTOCOMMIT_INT_CONFIG	INTEGER	-	50	O	It specifies how often the consumer commits its current position, which can be useful for ensuring message processing progress.
KAFKA_COMMIT_INT_CONFIG	INTEGER	-	50	O	this property will configure the interval at which Kafka consumer commits offsets.
KAFKA_NUMBER_THREADS_CONFIG	INTEGER	-	6	O	this property is used to configure the number of threads or consumers that Kafka Streams or Kafka consumers will use for processing messages
KAFKA_MAX_AGE_CONFIG	INTEGER	-	7500	O	This property will be used to set a maximum age for Kafka consumer records
KAFKA_CONSUMER_STRATEGY	STRING	-	org.apache.kafka.clients.consumer.RoundRobinAssignor	O	This property will be used to the strategy used for partition assignment when consuming messages from Kafka topics

Table 3-8 (Cont.) Correlation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
PRODUCERS_ACKNOWLEDGMENTS	INTEGER	-	0	O	producer acknowledgments
CONSUMER_POLL_MS	INTEGER	-	50	O	Polling time in ms for consumer
BATCH_SIZE	INTEGER	-	65536	O	The maximum amount of data to be collected before sending the batch.
LINGER_MS	INTEGER	-	1	O	The time to wait before sending messages out to Kafka
REQUEST_TIMEOUT_MS	INTEGER	-	1000	O	The configuration controls the maximum amount of time the client will wait for the response of a request
OCNADD_KAFKA_SSL_PROTOCOL	STRING	-	TLSv1.3	O	SSL Protocol version
OCNADD_KAFKA_SECURITY_PROTOCOL_SASL	STRING	-	SASL_SSL	O	describes SASL_SSL kafka security Protocol
OCNADD_KAFKA_SECURITY_PROTOCOL_SSL	STRING	-	SSL	O	describes SSL kafka security Protocol
OCNADD_KAFKA_SASL_MECHANISM	STRING	-	PLAIN	O	describes SASL SCRAM mechanism
OCNADD_KAFKA_SASL_JAAS_USERNAME	STRING	-	ocnadd	O	kafka default jaas username present
OCNADD_KAFKA_SASL_JAAS_MODULE	STRING	-	org.apache.kafka.common.security.plain.PlainLoginModule	O	kafka Login module
OCNADD_KAFKA_JAAS_SECRET_KEY	STRING	-	jaas_password	O	jaas password for kafka authentication taken from the jaas-secret with this key
OCNADD_KAFKA_JASS_SECRET_NAME	STRING	-	jaas-secret	O	jaas-secret name
CORRELATION_LOG_LEVEL_KAFKA	STRING	[ON,OFF]	OFF	O	Kafka Streams Log Level

Table 3-8 (Cont.) Correlation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
OCNADD_CORRELATION_LOG_ROOT	STRING	-	WARN	O	root log level
OCNADD_CORRELATION_LOG_NETTY	STRING	-	INFO	O	netty log level
KAFKASTREAMS_PUNCTUATOR_TIMER	INTEGER	-	2000	O	Kafka Stream Scheduler Timer to loop through the Local cache for Timer Expiry Scenario
Environmental variables for Diameter Correlation service are declared under "ocnaddadminsvc.diametercorrelation.env" section **(Note:** The Kafka parameters for Diameter correlation are same as explained in the ocnaddadminsvc.correlation.env)					
diametercorrelation.resources.limits.cpu	INTEGER	-	3	M	Number of maximum CPUs for each Diameter Correlation instance
diametercorrelation.resources.limits.memory	STRING	-	64Gi	M	Max Memory limit for each Diameter Correlation instance
diametercorrelation.resources.limits.ephemeralstorage	STRING	-	800Mi	M	Ephemeral Storage for each Diameter Correlation instance
diametercorrelation.resources.requests.cpu	INTEGER	-	3	M	Minimum number of CPUs required for each Diameter Correlation instance
diametercorrelation.resources.requests.memory	STRING	-	24Gi	M	minimum Memory required for each Diameter Correlation instance
diametercorrelation.resources.requests.ephemeralstorage	STRING	-	400Mi	M	minimum Ephemeral Storage required for each Diameter Correlation instance
diametercorrelation.minReplicas	INTEGER	-	1	M	Minimum replicas of Diameter correlation service instance

Table 3-8 (Cont.) Correlation Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
diametercorrelation.maxReplicas	INTEGER	-	4	M	Maximum replicas of Diameter correlation service instance

3.7.2 Storage Adapter Service Parameters

All the Storage Adapter Service parameters are available under `ocnaddadminsvc.storageadapter` section in `ocnadd-mediation-custom-values.yaml` file.

Table 3-9 Storage Adapter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
resources.limits.cpu	INTEGER	-	3	M	Number of maximum CPUs for each storage adapter instance
resources.limits.memory	STRING	-	64Gi	M	Max Memory limit for each storage adapter instance
resources.limits.ephemeralstorage	STRING	-	400Mi	M	Ephemeral Storage for each storage adapter instance
resources.requests.cpu	INTEGER	-	3	M	Minimum number of CPUs required for each storage adapter instance
resources.requests.memory	STRING	-	64Gi	M	minimum Memory required for each storage adapter instance
resources.requests.ephemeralstorage	STRING	-	400Mi	M	minimum Ephemeral Storage required for each storage adapter instance
Environmental variables for Storage Adapter service are declared under "ocnaddadminsvc.storageadapter.env" section					
OCNADD_STORAGE_ADAPTER_HTTP2_ENABLED	BOOLEAN	[true,false]	true	M	The flag to indicate if HTTP2 should be used or not. Default is true
KAFKA_SOCKET_BYTES_BUFFER	INTEGER	-	104857	O	Kafka Socket Buffer setting for consumer
STORAGE_ADAPTER_KAFKA_FETCH_MIN_BYTES	INTEGER	-	1	O	The minimum amount of data per-partition the server will return
STORAGE_ADAPTER_KAFKA_MAX_PARTITION_FETCH_BYTES	INTEGER	-	104858	O	The maximum amount of data per-partition the server will return.

Table 3-9 (Cont.) Storage Adapter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
STORAGE_ADAPTER_KAFKA_FETCH_MAX_WAIT_MS	INTEGER	-	100	O	The maximum amount of time the server will block before answering the fetch request if there isn't sufficient data to immediately satisfy the requirement given by fetch.min.bytes
STORAGE_ADAPTER_KAFKA_SESSION_TIME_OUT	INTEGER	-	90000	O	The timeout used to detect client failures when using Kafka's group management facility.
STORAGE_ADAPTER_KAFKA_HEARTBEAT_INTERVAL_MS	INTEGER	-	30000	O	The expected time between heartbeats to the consumer coordinator when using Kafka's group management facilities
STORAGE_ADAPTER_KAFKA_MAX_POLL_INTERVAL_MS	INTEGER	-	240000	O	The maximum delay between invocations of poll() when using consumer group management
STORAGE_ADAPTER_KAFKA_MAX_POLL_RECORDS	INTEGER	-	900	O	The maximum number of records returned in a single call to poll()
STORAGE_ADAPTER_KAFKA_OFFSET_CONFIG	STRING	-	latest	O	default kafka data stream offset config
STORAGE_ADAPTER_KAFKA_NUM_THREADS_CONFIG	INTEGER	-	6	O	this property is used to configure the number of threads or consumers that Kafka Streams or Kafka consumers will use for processing messages
PRODUCERS_ACKNOWLEDGMENTS	INTEGER	-	0	O	producer acknowledgments
STORAGE_ADAPTER_KAFKA_CONSUMER_POLL_MS	INTEGER	-	50	O	Polling time in ms for consumer
KAFKA_BATCH_SIZE	INTEGER	-	75000	O	The maximum amount of data to be collected before sending the batch.
STORAGE_ADAPTER_LOG_LEVEL_KAFKA	STRING	-	OFF	O	Kafka log level
OCNADD_STORAGE_ADAPTER_LOG_ROOT	STRING	-	WARN	O	root log level
OCNADD_STORAGE_ADAPTER_LOG_NETTY	STRING	-	INFO	O	netty log level
OCNADD_INTERNAL_CLIENT_SSL_PROTOCOL	STRING	-	TLS	O	The secure protocol used between Storage adapter and internal DD services for the HTTP communication

Table 3-9 (Cont.) Storage Adapter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
OCNADD_INTERNAL_CLIENT_SSL_PROTOCOLS	STRING	-	TLSv1.2, TLSv1.3	O	The TLS version supported by HTTP client in the storage adapter service
OCNADD_INTERNAL_CLIENT_SSL_HANDSHAKE_TIMEOUT	STRING	-	30s	O	The SSL handshake timeout value in the HTTP client used in the storage adapter service
STORAGE_ADAPTER_METRICS_ENABLED	BOOLEAN	true/ false	false	O	Parameter to enable the metrics pegging for the storage adapter, defaule is false
STORAGE_ADAPTER_EVENT_ENABLED	BOOLEAN	true/ false	true	O	Parameter to enable the events on the storage adapter, default is true
EVENT_DELETE_BATCH_SIZE	INTEGER	-	5000	O	The size of the event list that can be deleted by storage adapter, default is 5000

3.7.3 Ingress Adapter Service Parameters

All the Ingress adapter service parameters are available under `ocnaddadminsvc.ingressadapter` section in `ocnadd-mediation-custom-values.yaml` file of current release.

Table 3-10 Ingress Adapter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
resources.limits.cpu	INTEGER	-	3	M	Number of maximum CPUs for each ingress adapter instance
resources.limits.memory	STRING	-	64Gi	M	Max Memory limit for each ingress adapter instance
resources.limits.ephemeralstorage	STRING	-	400Mi	M	Ephemeral Storage for each ingress adapter instance
resources.requests.cpu	INTEGER	-	3	M	Minimum number of CPUs required for each ingress adapter instance
resources.requests.memory	STRING	-	64Gi	M	minimum Memory required for each ingress adapter instance
resources.requests.ephemeralstorage	STRING	-	400Mi	M	minimum Ephemeral Storage required for each ingress adapter instance
Environmental variables for Ingress Adapter service are declared under "ocnaddadminsvc.ingressadapter.env" section					

Table 3-10 (Cont.) Ingress Adapter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
INGRESS_ADAPTER_LOG_LEVEL_ROOT	STRING	-	OFF	O	Kafka log level
INGRESS_ADAPTER_NETTY_LOG_LEVEL	STRING	-	WARN	O	root log level
INGRESS_ADAPTER_LOG_LEVEL_KAFKA	STRING	-	INFO	O	netty log level
INGRESS_ADAPTER_KEYSTORE_TYPE	STRING	-	PKCS12	M	trust store client key Type
INGRESS_ADAPTER_TRUSTSTORE_TYPE	STRING	-	PKCS12	M	trust store key Type
INGRESS_HTTPSERVER_ROUTE_PATH	STRING	-	/ocnadd-nonoraclenf/v1/streaming	M	The URL at which the client should stream the data towards ingress adapter
INGRESS_HTTPSERVER_READ_TIMEOUT_MS	INTEGER	-	30000	O	The ingress adapter read timeout in milisec
INGRESS_HTTPSERVER_REQUEST_TIMEOUT_MS	INTEGER	-	30000	O	The ingress adapter request timeout in milisec
INGRESS_HTTPSERVER_CONNECT_TIMEOUT_MS	INTEGER	-	60000	O	The ingress adapter connect timeout in milisec
INGRESS_HTTPSERVER_IDLE_TIMEOUT_MS	INTEGER	-	120000	O	The ingress adapter idle timeout in milisec
INGRESS_HTTPSERVER_SOCKET_RECEIVE_BUF	INTEGER	-	10485	O	The socket receive buffer size
INGRESS_HTTPSERVER_SOCKET_RECEIVE_BUF_PORTION	INTEGER	-	100	O	The socket receive buffer size multiple factor. the actual read buffer bytes will be (INGRESS_HTTPSERVER_SOCKET_RECEIVE_BUF * INGRESS_HTTPSERVER_SOCKET_RECEIVE_BUF_PORTION)
INGRESS_HTTPSERVER_SOCKET_TIMEOUT_MS	INTEGER	-	60000	O	The ingress adapter socket timeout in milisec

Table 3-10 (Cont.) Ingress Adapter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
INGRESS_HTTPSERVER_SOCKET_KEEPALIVE	BOOLEAN	[true,false]	true	O	The flag to indicate if keepalive should be used in the connection
INGRESS_HTTPSERVER_CLOSE_NOTIFY_FLUSH_TIIMEOUT_MS	INTEGER	-	30000	O	Notification flush timeout in milisec
INGRESS_HTTPSERVER_CLOSE_NOTIFY_READ_TIIMEOUT_MS	INTEGER	-	30000	O	Notification read timeout in milisec
INGRESS_HTTPSERVER_SSL_HANDSHAKE_TIMEOUT_MS	INTEGER	-	30000	O	SSL handshake timeout in milisec
KAFKA_SECURITY_PROTOCOL	STRING	-	PLAINTEXT	O	describes kafka security Protocol
INGRESS_ADAPTER_SECURITY_PROTOCOL	STRING	-	SSL	O	describes ingress adapter security Protocol
KAFKA_SSL_PROTOCOL	STRING	-	TLSv1.2	O	SSL Protocol version
KAFKA_SASL_ENABLED	BOOLEAN	[true,false]	false	O	The flag to indicate if SASL is used for the authentication
KAFKA_JAAS_CONFIG_MODULE	STRING	-	org.apache.kafka.common.security.plain.PlainLoginModule	O	kafka Login module
KAFKA_JAAS_CONFIG_USER	STRING	-	username	O	kafka default jaas username present
KAFKA_JAAS_CONFIG_PASS	STRING	-	secret	O	kafka default jaas password present
KAFKA_SASL_MECHANISM	STRING	-	PLAIN	O	describes SASL SCRAM mechanism
externalAccess.enabled	BOOLEAN	[true,false]	false	O	The flag to indicate if external access is enabled for the ingress adapter
externalAccess.staticLoadBalancerIp	STRING	-	10.10.10.1	O	Default static loadbalancer IP address

3.8 Kafka Configuration Parameters

Applicable to both Relay Agent and Mediation Kafka Clusters.

Table 3-11 Kafka Configuration Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
kafkaBroker.kafkaProperties.logdirs	String	-	/kafka/logdir/kafka-logs	M	The path to store the kafka logs
kafkaBroker.replicas	Int		4		The number of replicas that should be available for the pod.
kafkaBroker.pvcClaimSize	String		10Gi	M	Size of Block Volume to attach to kafka.
target.averageCpuUtilPercentage	Int		50		The target average CPU utilization percentage.
target.memoryUtilPercentage	Int		80		The target average memory utilization percentage.
kafkaBroker.resource.limits.cpu	Int		5		The maximum limit for the number of CPUs used for the container.
kafkaBroker.resource.limits.memory	String		24Gi		The maximum limit for the size of the memory used for the container.
kafkaBroker.kafkaProperties.logRetentionMinutes	Int		5	M	Log Retention Time of Topic Data in Minutes.
kafkaBroker.kafkaProperties.logCleanerDeleteRetentionMs	String		2340000	M	The amount of time to retain tombstone message markers for log compacted topics.
kafkaBroker.kafkaProperties.kafkaSslProtocol	String		TLSv1.2, TLSv1.3	M	TLS version supported.
kafkaBroker.kafkaProperties.socketSendBufferBytes	Int		1048576 0	M	TCP socket buffer sizes for the producer.
kafkaBroker.kafkaProperties.socketReceiveBufferBytes	Int		1048576 0	M	TCP socket buffer sizes for the consumer.
kafkaBroker.kafkaProperties.socketRequestMaxBytes	Int		1048576 00	M	The maximum number of bytes in a socket request.
kafkaBroker.kafkaProperties.queuedMaxRequests	Int		4096	M	Number of concurrent connections.
kafkaBroker.kafkaProperties.numIoThreads	Int		820	M	Number of threads that pick up requests from the request queue to process them.
kafkaBroker.kafkaProperties.numNetworkThreads	Int		820	M	Network threads handle requests to the Kafka cluster, such as produce and fetch requests from client applications.

Table 3-11 (Cont.) Kafka Configuration Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
kafkaBroker.kafkaProperties.numReplicaFetchers	Int		640	M	Number of fetcher threads used to replicate records from each source broker.
kafkaBroker.kafkaProperties.backgroundThreads	Int		256	M	The number of threads to use for various background processing tasks.
kafkaBroker.kafkaProperties.replicaFetchMinBytes	Int		619200	M	Minimum bytes expected for each fetch response.
kafkaBroker.kafkaProperties.replicaFetchMaxBytes	Int		37152000	M	The maximum number of bytes we will return for a fetch request.
kafkaBroker.kafkaProperties.replicaFetchWaitMaxMs	Int		500	M	The maximum wait time for each fetcher request issued by follower replicas.
kafkaBroker.kafkaProperties.replicaSocketReceiveBufferBytes	Int		10485760	M	The socket receive buffer for network requests.
kafkaBroker.kafkaProperties.offsetsTopicReplicationFactor	Int		3	M	The replication factor for the offsets topic (set higher to ensure availability). Internal topic creation will fail until the cluster size meets this replication factor requirement.
kafkaBroker.kafkaProperties.transactionStateLogReplicationFactor	Int		3	M	The replication factor for the transaction topic (set higher to ensure availability). Internal topic creation will fail until the cluster size meets this replication factor requirement.
kafkaBroker.externalAccess.enabled	Boolean		false	M	Flag to enable External access for Kafka.
kafkaBroker.externalAccess.autoDiscovery	Boolean		false	M	Flag to enable auto-discovery of LoadBalancer IPs.
kafkaBroker.externalAccess.type	String		LoadBalancer	M	Service Type of Kafka Broker.
kafkaBroker.externalAccess.setstaticLoadBalancerIps	Boolean		false	M	Setting Static LoadBalancer IPs.
kafkaBroker.externalAccess.LoadBalancerIPList	List		[ ]	C	List if LoadBalancer Static IP available for use.

Table 3-11 (Cont.) Kafka Configuration Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
kafkaBroker.kafkaProperties.ramDriveStorage	Boolean	true/ false	Relay Agent Kafka: true Mediation Kafka: false	C	The property is used to enable the RAM based storage for the Kafka cluster in the worker group. When enabled the messages in the Kafka topic will be stored inside the RAM and a very minumum retention will be available. The default value is false. By Default, the Relay Agent Kafka cluster will use RAM drive as storage where as Mediation Kafka cluster will be CEPH based persistence storage

3.9 UI Router Parameters

Listed below are the UI Router Parameters:

Table 3-12 UI Router Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
DD_UIAPI	STRING	-	http://ocnaddgui:80	M	The target endpoint of ocnaddgui service and use to configure the GUI.
groupNamePromIpConfig	OBJECT	-	-	M	List of all the OCNADD group names and their corresponding prometheus IPs to route the GUI request and forward the response
compartmentId	STRING	-	ocid.compartment.oc1.xxxxx	C	Compartment ID required on OCI
disk_namespace_oci	STRING	-	oci_computeagent	C	Disk namespace required on OCI
namespace_oci	STRING	-	ocnaddgui_oci_metrics	C	The OCI namespace
DD_PROMETHEUS_PATH	STRING		/cluster_name/prometheus/api/v1/query_range	M	The Prometheus endpoint API URL path. The update for cluster name will be automatically managed by the application. User do not need to modify this parameter.
resources.limits.cpu	INTEGER	-	1	M	Number of maximum CPUs for each UI router service instance

Table 3-12 (Cont.) UI Router Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
resources.limits.memory	STRING	-	1Gi	M	Max Memory limit for each UI router service instance
resources.limits.ephemeralstorage	STRING	-	500Mi	M	Ephemeral Storage for each UI router service
resources.requests.cpu	INTEGER	-	1	M	Minimum number of CPUs required for each UI router service instance
resources.requests.memory	STRING	-	1Gi	M	Minimum Memory required for each UI router instance
resources.requests.ephemeralstorage	STRING	-	100Mi	M	Minimum Ephemeral Storage required for each UI router instance

3.10 Filter Service Parameters

Filter Service Parameters are present under `ocnaddfilter` section in `ocnadd-mediation-custom-values.yaml` file of current release.

Table 3-13 Filter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
minReplicas	INTEGER	-	1	M	minimum number of Replicas of Filter Service
maxReplicas	INTEGER	-	3	M	maximum number of Replicas of Filter Service
resources.limits.cpu	INTEGER	-	2	M	Number of maximum CPUs for each Filter service instance
resources.limits.memory	String	-	3Gi	M	Max Memory limit for each Filter service instance
resources.limits.ephemeralstorage	String	-	800Mi	M	Ephemeral Storage for each Filter service instance
resources.requests.cpu	INTEGER	-	2	M	Minimum number of CPUs required for each Filter service instance
resources.requests.memory	String	-	3Gi	M	minimum Memory required for each Filter service instance
resources.requests.ephemeralstorage	String	-	500Mi	M	minimum Ephemeral Storage required for each Filter service instance
Environmental variables are present under section "ocnaddfilter.env"					

Table 3-13 (Cont.) Filter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ENABLE_FILTER_METRICS	BOOLEAN	[true/ false]	true	O	To enable/disable filter metrics, default is true
OCNADD_KAFKA_SSL_PROTOCOL	STRING	-	TLSv1.3	O	kafka SSL protocol version
KAFKA_PRODUCER_SSL_CLIENT_AUTH	BOOLEAN	[true/ false]	false	O	whether kafka producer client auth is required or not
KAFKA_MAX_AGE_CONFIG	INTEGER	-	7500	O	The period of time in milliseconds after which we force a refresh of metadata.
KAFKA_FETCH_MIN_BYTES	INTEGER	-	1	O	The minimum amount of data per-partition the server will return
KAFKA_FETCH_MAX_BYTES	STRING	-	57672000	O	The maximum amount of data per-partition the server will return
KAFKA_MAX_PARTITIONS_FETCH_BYTES	STRING	-	1048580	O	The maximum amount of data per-partition the server will return.
FETCH_MAX_WAIT_MS	INTEGER	-	100	O	The maximum amount of time the server will block before answering the fetch request if there isn't sufficient data to immediately satisfy the requirement given by fetch.min.bytes
SESSION_TIME_OUT	INTEGER	-	15000	O	The timeout used to detect client failures when using Kafka's group management facility.
HEARTBEAT_INTERVAL_MS	INTEGER	-	5000	O	The expected time between heartbeats to the consumer coordinator when using Kafka's group management facilities
MAX_POLL_INTERVAL_MS	INTEGER	-	240000	O	The maximum delay between invocations of poll() when using consumer group management
MAX_POLL_RECORDS	INTEGER	-	1500	O	The maximum number of records returned in a single call to poll()
CONSUMER_POLL_MS	INTEGER	-	50	O	Polling time in ms for consumer
PRODUCERS_ACKNOWLEDGMENTS	INTEGER	-	1	O	The number of acknowledgments the producer requires the leader to have received before considering a request complete
BATCH_SIZE	INTEGER	-	130000	O	The maximum amount of data to be collected before sending the batch.
LINGER_MS	INTEGER	-	2	O	The time to wait before sending messages out to Kafka

Table 3-13 (Cont.) Filter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
REQUEST_TIMEOUT_MS	INTEGER	-	1000	O	The configuration controls the maximum amount of time the client will wait for the response of a request
TRANSACTION_FILTER	BOOLEAN	[true/ false]	true	O	To enable or disable transaction filtering
KAFKA_SOCKET_BYTES_BUFFER	STRING	-	1048570	O	Kafka Socket Buffer setting for consumer
FILTER_KAFKA_PARTITIONER_STRATEGY	STRING	[key/ custom/ roundrobin]	key	O	Kafka record partitioner strategy
OCNADD_FILTER_KAFKA_TOPIC_INITIAL_DELAY	STRING	-	10s	O	The parameter denotes the initial delay in checking for the Kafka topic existence, default is 10sec
OCNADD_FILTER_KAFKA_TOPIC_RETRY_THRESHOLD	STRING	-	20s	O	The parameter denotes the retry interval for checking the Kafka topic existence, default is 20sec
OCNADD_CONFIG_RETRY_COUNT	INTEGER	-	3	O	The number of retries for the communication towards the configuration service in case of failure, default is 3
OCNADD_CONFIG_RETRY_DELAY_MS	INTEGER	-	5000	O	The retry interval for the communication towards the configuration service in case of failure, default is 5ms
KAFKA_DESCRIBE_TOPIC_TIMEOUT_MS	INTEGER	-	10000	O	The timeout for the request to get the Kafka describe topic output from Kafka cluster, default is 10ms
OCNADD_FILTER_HEALTH_SVC_TYPE	STRING	-	FILTER_SERVICE	M	The type with which filter service registers with the health monitoring service
OCNADD_FILTER_HEALTH_HB_TIMER	INTEGER	-	10000	M	The heartbeat timer on the filter service to exchange the heartbeat with the health monitoring service, default is 10sec
OCNADD_FILTER_HEALTH_RETRY_COUNT	INTEGER	-	1	M	The number of retries with the healt monitoring service for the registration of filter service health profile, default is 1
OCNADD_FILTER_HEALTH_RETRY_DELAY	INTEGER	-	2	M	The retry delay between two consecutive retries for the health profile registration of filter service, default is 2sec

Table 3-13 (Cont.) Filter Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
OCNADD_FILTER_MAX_REPLICA	INTEGER	-	1	M	The number of maximum replicas, the health service instance reports to the health service during health profile registration.
OCNADD_KAFKA_SSL_PROTOCOL	STRING	-	TLSv1.3	O	SSL Protocol version
OCNADD_KAFKA_SECURITY_PROTOCOL_SSL	STRING	-	SSL	O	The Kafka security protocol for the filter service for SSL connection with Kafka

3.11 Redundancy Agent Service Parameters

Table 3-14 Redundancy Agent Service Parameter

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
autoScaling.enabled	BOOLEAN	true/ false	false	M	Allow `HorizontalAutoScalar` of `ocnaddredundancy` pods.
minReplicas	INTEGER	-	1	C	Number of minimum replicas for HPA.
maxReplicas	INTEGER	-	1	C	Number of maximum replicas for HPA.
resources.limit.cpu	INTEGER	-	2	M	Maximum number of CPU for each pod.
resources.limits.memory	STRING		1Gi	M	Maximum memory limit for each service instance.
resources.limits.ephemeralstorage	STRING		500Mi	M	Ephemeral storage for each service instance.
resources.requests.cpu	INTEGER		2	M	Minimum number of CPUs required for each service instance.
resources.requests.memory	STRING		1Gi	M	Minimum memory required for each service instance.
resources.requests.ephemeralstorage	STRING		500Mi	M	Minimum ephemeral storage required for each service instance.
resources.target.averageCpuUtilsPercentage	INTEGER	-	85	C	Threshold set for Pod AutoScalar.
Environmental variables are present under section ocnaddredundancyagent.env					

Table 3-14 (Cont.) Redundancy Agent Service Parameter

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
OCNADD_REDUNDANCY_HB_INTERVAL	INTEGER	-	10	O	Interval of heartbeat requests sent to primary agent by secondary agent.
OCNADD_REDUNDANCY_HB_MISSING	INTEGER		3	O	Max unsuccessful heartbeat in case of secondary agent or max missing heartbeat in case of primary agent.
OCNADD_REDUNDANCY_KAFKA_DELAY_MS	INTEGER		2000	O	Delay before starting periodic Kafka ingress traffic rate.
OCNADD_REDUNDANCY_KAFKA_INTERVAL_MS	INTEGER		500	O	Interval of periodic Kafka ingress traffic check, will switch mode of secondary agent if change is required during the check.
OCNADD_REDUNDANCY_HEALTH_RETRY_COUNT	INTEGER		10	O	Number of retries for Health registration.
OCNADD_REDUNDANCY_HEALTH_RETRY_DELAY	INTEGER		15	O	Delay between each retries for Health Registration.
OCNADD_REDUNDANCY_HEALTH_HB_TIMER	INTEGER		120000	O	Heart Beat Timer interval to health monitoring service.
OCNADD_REDUNDANCY_HEALTH_SVC_TYPE	STRING		REDUNDANCY	O	Health Registration name for REDUNDANCY agent.

3.12 Export Service Parameters

Table 3-15 Export Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
autoScaling.enabled	BOOLEAN	true/ false	true	M	Allow HorizontalAutoScalar of ocnaddexport service pods.
minReplicas	INTEGER	-	1	C	Number of minimum replicas for HPA.
maxReplicas	INTEGER	-	2	C	Number of maximum replicas for HPA.
resources.limit.cpu	INTEGER	-	6	M	Max number of cpu for each pod.

Table 3-15 (Cont.) Export Service Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
resources.limits.memory	STRING	-	24Gi	M	Max memory limit for each service instance.
resources.limits.ephemeralstorage	STRING	-	2Gi	M	Ephemeral storage for each service instance.
resources.requests.cpu	INTEGER	-	4	M	Minimum number of CPUs required for each service instance.
resources.requests.memory	STRING	-	4Gi	M	Minimum memory required for each service instance.
resources.requests.ephemeralstorage	STRING	-	100Mi	M	Minimum ephemeral storage required for each service instance.
Environmental variables are present under section ocnaddexport.env					
EXPORT_BLOCKINGQUEUE_SIZE	INTEGER	-	10	O	The queue size to store the result set from the database for the export.
EXPORT_SEQUENCING	BOOLEAN	true,false	true	O	The parameter to decide if the result set needs to be in sequence based on the record timestamp or not.

3.13 Helm Parameter Configuration for OCCM

Table 3-16 Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
global.certificates.occm.enabled	BOOLEAN	true/ false	false	M	Whether to use OCCM for creating services.
global.certificates.occm.issuer	STRING	-	CA1	M	Name of the Issuer configured in OCCM to use to create certificate
global.certificates.occm.renewBefore	INTEGER	-	14	M	Number of days before expiry,before which OCCM will automatically update the certificates
global.certificates.occm.days	INTEGER	-	90	M	Number of days for which certificates will be valid
global.certificates.occm.cncc.cncc_iam_ingress_gateway.external_ip	STRING	-	-	M	Load balancer IP address of CNCC IAM Ingress Gateway Service
global.certificates.occm.cncc.cncc_iam_ingress_gateway.port	INTEGER	-	80	M	Port of CNCC IAM Ingress Gateway Service

Table 3-16 (Cont.) Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
global.certificates.occm.cncc.cncc_mcore_ingress_gateway.external_ip	STRING	-	-	M	Load balancer IP address of CNCC MCORE Ingress Gateway Service
global.certificates.occm.cncc.cncc_mcore_ingress_gateway.port	INTEGER	-	80	M	Port of CNCC MCORE Ingress Gateway Service
global.certificates.occm.cncc.ccncId	STRING	-	Cluster1	M	ID of CNCC owner of OCCM instance
global.certificates.occm.cncc.occm_cncc_instance_id	STRING	-	Cluster1-occm-instance1	M	OCCM instance id
global.certificates.occm.subject.country	STRING	-	-	M	Specify the country field (C) in DN for each certificate
global.certificates.occm.subject.state	STRING	-	-	M	Specify the state field (S) in DN for each certificate
global.certificates.occm.subject.location	STRING	-	-	M	Specify the location field (L) in DN for each certificate
global.certificates.occm.subject.organization	STRING	-	-	M	Specify the organization field (O) in DN for each certificate
global.certificates.occm.subject.country.organizationUnit	STRING	-	-	M	Specify the organization unit field (OU) in DN for each certificate
global.certificates.occm.occm_cacert	STRING	-	occm-ca-secret	O	Name of the Secret storing CA certificate/certificate chain.
global.certificates.occm.truststore_keystore_secret	STRING	-	occm-truststore-keystore-secret	O	Name of the Secret storing truststore and keystore key
global.certificates.occm.occm_secret	STRING	-	occm-secret	O	Name of the Secret storing CNCC user credentials
global.certificates.occm.volumes.json	STRING	-	/occm-request	O	Mount path of the JSONs used when sending request to OCCM
global.certificates.occm.volumes.script	STRING	-	/occm-script	O	Mount path of the script used to send request to OCCM
global.certificates.occm.keyAlgorithm	STRING	RSA/EC	RSA	C	Select OCCM key algorithm, RSA for RSA based key generation and EC for ECDSA based key generation

Table 3-16 (Cont.) Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
global.certificates.occm.keySize	STRING	KEYSIZE_2048/ KEYSIZE_4096	KEYSIZE_2048	C	Defines the keySize of RSA based key generation
global.certificates.occm.ecCurve	STRING	SECP384r1/ SECP256r1	SECP384r1	C	Define the curve parameter when keyAlgorithm select is EC
global.ocnaddmanagement.certificates.occm.san.redundancy_agent.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for Redundancy Agent certificates is required. Should be enabled post-installation when two site redundancy is enabled.
global.ocnaddmanagement.certificates.occm.san.redundancy_agent.ips	LIST[STRING]	-	["10.10.10.10"]	C	IPs to add in SAN for Redundancy Agent certificate. Provide the Load balancer IP during installation if static IP for loadbalancer is chosen
global.ocnaddmanagement.certificates.occm.san.redundancy_agent.uuid.server	STRING	-	-	C	UUID of existing Redundancy Agent certificate with names prefixed by REDUNDANCYAGENT-SECRET-SERVER
global.ocnaddmanagement.certificates.occm.san.management_gateway.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for Management Gateway certificates is required.
global.ocnaddmanagement.certificates.occm.san.management_gateway.ips	LIST[STRING]	-	["10.10.10.10"]	C	IPs to add in SAN for Management Gateway certificate. Provide the Load balancer IP during installation if static IP for loadbalancer is chosen
global.*ocnaddmanagement*.certificates.occm.san.management_gateway.uuid.server	STRING	-	-	C	UUID of existing Management Gateway certificate with names prefixed by MANAGEMENTGATEWAY-SECRET-SERVER
global.ocnaddrelayagent.certificates.occm.san.kafka.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for Relay agent Kafka certificates is required. Should be enabled post-installation when external access of Kafka is required.
global.ocnaddrelayagent.certificates.occm.san.kafka.ips	LIST[STRING]	-	["10.10.10.10", "10.10.10.11", "10.10.10.12"]	C	IPs to add in SAN for Relay agent Kafka certificate. Provide the Load balancer IPs during installation if static IPs for loadbalancer are chosen

Table 3-16 (Cont.) Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
global.ocnaddrelayagent.certificates.occm.san.kafka.uuid.server	STRING	-	-	C	UUID of existing Relay agent Kafka broker certificate with names prefixed by KAFKABROKER-SECRET-SERVER
global.ocnaddrelayagent.certificates.occm.san.relay_gateway.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for Relay Agent Gateway certificates is required.
global.ocnaddrelayagent.certificates.occm.san.relay_gateway.ips	LIST[STRING]	-	["10.10.10.10"]	C	IPs to add in SAN for Relay Agent Gateway certificate. Provide the Load balancer IP during installation if static IP for loadbalancer is chosen
global.ocnaddrelayagent.certificates.occm.san.relay_gateway.uuid.server	STRING	-	-	C	UUID of existing Relay Agent Gateway certificate with names prefixed by RELAYAGENTGATEWAY-SECRET-SERVER
global.ocnaddrelayagent.certificates.occm.san.vcollector.enabled	BOOLEAN	true/ false	false	C	Enable this property when vCollector is enabled
global.ocnaddrelayagent.certificates.occm.san.vcollector.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for vCollector certificates is required.
global.ocnaddrelayagent.certificates.occm.san.vcollector.ips	LIST[STRING]	-	["10.10.10.10"]	C	IPs to add in SAN for vCollector certificate. Provide the Load balancer IP during installation if static IP for loadbalancer is chosen
global.ocnaddrelayagent.certificates.occm.san.vcollector.uuid.server	STRING	-	-	C	UUID of existing vCollector certificate with names prefixed by VCOLLECTOR-SECRET-SERVER
global.ocnaddmediation.certificates.occm.san.kafka.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for Mediation Kafka certificates is required. Should be enabled post-installation when external access of Kafka is required.
global.ocnaddmediation.certificates.occm.san.kafka.ips	LIST[STRING]	-	["10.10.10.10", "10.10.10.11", "10.10.10.12"]	C	IPs to add in SAN for Mediation Kafka certificate. Provide the Load balancer IPs during installation if static IPs for loadbalancer are chosen
global.ocnaddmediation.certificates.occm.san.kafka.uuid.server	STRING	-	-	C	UUID of existing Mediation Kafka broker certificate with names prefixed by KAFKABROKER-SECRET-SERVER

Table 3-16 (Cont.) Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
global.ocnaddmediation.certificates.occm.san.ingress_adapter.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for Ingress Adapter certificates is required. Should be enabled post-installation when external access to Ingress Adapter/s is needed.
global.ocnaddmediation.certificates.occm.san.ingress_adapter.ips	LIST[STRING]	-	["10.10.10.10"]	C	IPs to add in SAN for Ingress Adapter certificate
global.ocnaddmediation.certificates.occm.san.ingress_adapter.uuid.server	STRING	-	-	C	UUID of existing Ingress Adapter certificate with names prefixed by INGRESSADAPTER-SECRET-SERVER
global.ocnaddmediation.certificates.occm.san.mediation_gateway.update_required	BOOLEAN	true/ false	false	C	If update of SAN field for Mediation Gateway certificates is required.
global.ocnaddmediation.certificates.occm.san.mediation_gateway.ips	LIST[STRING]	-	["10.10.10.10"]	C	IPs to add in SAN for Mediation Gateway certificate. Provide the Load balancer IP during installation if static IP for loadbalancer is chosen
global.ocnaddmediation.certificates.occm.san.mediation_gateway.uuid.server	STRING	-	-	C	UUID of existing Mediation Gateway certificate with names prefixed by MEDIATIONGATEWAY-SECRET-SERVER

3.14 Helm Parameter Configuration for Network Policy

Table 3-17 Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmanagement.network.policy.enable	BOOLEAN	true/ false	false	M	Network Policy enable for intercommunication of OCNADD Management Group services
ocnaddmanagement.network.ingress.denyall	BOOLEAN	true/ false	true	C	Deny all other ingress traffic

Table 3-17 (Cont.) Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmanagement.network.ingress.alarm	BOOLEAN	true/ false	true	C	Allow ingress traffic for alarm service
ocnaddmanagement.network.ingress.config	BOOLEAN	true/ false	true	C	Allow ingress traffic for configuration service
ocnaddmanagement.network.ingress.health	BOOLEAN	true/ false	true	C	Allow ingress traffic for health monitoring service
ocnaddmanagement.network.ingress.agent	BOOLEAN	true/ false	true	C	Allow ingress traffic for Redundancy Agent service
ocnaddmanagement.network.ingress.export	BOOLEAN	true/ false	true	C	Allow ingress traffic for Export service
ocnaddmanagement.network.ingress.gateway	BOOLEAN	true/ false	true	C	Allow ingress traffic for Management Gateway service
ocnaddmanagement.network.ingress.namespaces.ocnaddgroups	OBJECT	NA	- ocnadd-relay - ocnadd-mediation	C	Network communication between allowed namespaces
ocnaddmanagement.network.ingress.namespaces.infra	OBJECT	NA	- occne-infra	C	Network communication between allowed Infra namespaces
ocnaddmanagement.network.ingress.external.enable	BOOLEAN	true/ false	false	C	Allow external network connections from configured IPs/CIDRs/Network
ocnaddmanagement.network.ingress.external.cidrs	OBJECT	NA	-	C	CIDRs for network communication
ocnaddrelayagent.network.policy.enable	BOOLEAN	true/ false	false	M	Network Policy enable for intercommunication of OCNADD Relay Agent Group services
ocnaddrelayagent.network.ingress.denyall	BOOLEAN	true/ false	true	C	Deny all other ingress traffic
ocnaddrelayagent.network.ingress.aggregation	BOOLEAN	true/ false	true	C	Allow ingress traffic for aggregation service
ocnaddrelayagent.network.ingress.diameteraggregation	BOOLEAN	true/ false	true	C	Allow ingress traffic for diameteraggregation service

Table 3-17 (Cont.) Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddrelayagent.network.ingress.kafka	BOOLEAN	true/ false	true	C	Allow ingress traffic for Kafka service
ocnaddrelayagent.network.ingress.gateway	BOOLEAN	true/ false	true	C	Allow ingress traffic for Gateway service
ocnaddrelayagent.network.ingress.namespaces.infra	OBJECT	NA	- occne-infra	C	Network communication between allowed Infra namespaces
ocnaddrelayagent.network.ingress.external.enable	BOOLEAN	true/ false	false	C	Allow external network connections from configured IPs/CIDRs/Network
ocnaddrelayagent.network.ingress.external.cidrs	OBJECT	NA	-	C	CIDRs for network communication
ocnaddmediation.network.policy.enable	BOOLEAN	true/ false	false	M	Network Policy enable for intercommunication of OCNADD Mediation Group services
ocnaddmediation.network.ingress.denyall	BOOLEAN	true/ false	true	C	Deny all other ingress traffic
ocnaddmediation.network.ingress.adapter	BOOLEAN	true/ false	true	C	Allow ingress traffic for adapter service
ocnaddmediation.network.ingress.admin	BOOLEAN	true/ false	true	C	Allow ingress traffic for admin service
ocnaddmediation.network.ingress.correlation	BOOLEAN	true/ false	true	C	Allow ingress traffic for correlation service
ocnaddmediation.network.ingress.diametercorrelation	BOOLEAN	true/ false	true	C	Allow ingress traffic for diametercorrelation service
ocnaddmediation.network.ingress.filter	BOOLEAN	true/ false	true	C	Allow ingress traffic for filter service
ocnaddmediation.network.ingress.kafka	BOOLEAN	true/ false	true	C	Allow ingress traffic for Kafka service
ocnaddmediation.network.ingress.gateway	BOOLEAN	true/ false	true	C	Allow ingress traffic for gateway service
ocnaddmediation.network.ingress.ingressadapter	BOOLEAN	true/ false	true	C	Allow ingress traffic for ingress adapter service

Table 3-17 (Cont.) Helm Parameter Configuration for OCCM

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
ocnaddmediation.network.ingress.storageAdapter	BOOLEAN	true/ false	true	C	Allow ingress traffic for storage adapter service
ocnaddmediation.network.ingress.namespaces.infra	OBJECT	NA	- occne-infra	C	Network communication between allowed Infra namespaces
ocnaddmediation.network.ingress.external.enable	BOOLEAN	true/ false	false	C	Allow external network connections from configured IPs/CIDRs/Network
ocnaddmediation.network.ingress.external.cidrs	OBJECT	NA	-	C	CIDRs for network communication

3.15 cnDBTier Customization Parameters

The Data Director uses cnDBTier as an independent database for the geo-redundant sites. Therefore, by default, the `ocnadd_dbtier_custom_values.yaml` provided with the OCNADD installation is for a single-site deployment of cnDBTier.

Single-site cnDBTier deployment mode: The georeplication is unavailable. Users must continue taking DB backups periodically, preferably on a daily basis, so that the same can be used when fault recovery scenarios arise. Refer to the section Fault Recovery for the backup options in the Data Director.

For information about the values of the following parameters, see the `ocnadd_dbtier_custom_values_25.2.200.yaml` file.

- Any change in the cnDBTier `custom_values` file introduced by the cnDBTier patch must be updated in the `custom_values` file provided by OCNADD before deployment.
- For detailed information on the cnDBTier resources, see the section *DB Profile* in the *Oracle Communications Network Analytics Data Director Benchmarking Guide*. The resources in the `ocnadd_dbtier_custom_values_25.2.200.yaml` should match the planning guide; if not, update them according to the planning guide.

The following table lists the customized cnDBTier parameters for OCNADD

Table 3-18 cnDBTier Customization Parameters

Parameter	Description	Version
global.repository	The value should be updated to point to the actual path of your docker registry respectively, for example occne-repo-host:5000/occne	24.3.0
global.sitename	This parameter must be set to the name of current cluster	24.3.0

Table 3-18 (Cont.) cnDBTier Customization Parameters

Parameter	Description	Version
global.domain	Set it to the name of Kubernetes cluster on which cnDbTier is installed, for example occne1-cgbu-cne-dbtier.	24.3.0
global.namespace	The Kubernetes namespace in which the cnDbTier is deployed	24.3.0
global.storageClassName	Storage class to be used. By default occne-dbtier-sc will be the storage class it can be changed to any storage class name which is currently configured in the cluster.	24.3.0
global.mgmReplicaCount	Default value to be used as in the file	24.3.0
global.ndbReplicaCount	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • Should be updated to 4 when cnDBTier is planned to be used as extended storage for xDRs • Default value (2) to be used in the file when cnDbTier is not used as extended storage	24.3.0
global.ndbappReplicaCount	Default value (2) to be used as in the file	24.3.0
global.ndbappReplicaMaxCount	Default value (4) to be used as in the file global.ndbappReplicaMaxCount should always be greater than global.ndbappReplicaCount	24.3.0
global.apiReplicaCount	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • In case of no replication, the minimum number of SQL nodes required is 0.	24.3.0
global.ndb.datamemory	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • Should be updated to 96G when cnDBTier is planned to be used as extended storage for xDRs • Default value (1G) to be used in the file when cnDbTier is not used as extended storage	24.3.0
global.mgm.ndbdisksize	Default value (30Gi) to be used as in the file	24.3.0
global.ndb.ndbdisksize	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • Should be updated to ndb.resources.limits.memory + 30Gi when cnDBTier is planned to be used as extended storage for xDRs • Default value (30Gi) to be used in the file when cnDbTier is not used as extended storage	24.3.0
global.ndb.ndbbackupdisksize	Default value (30Gi) to be used as in the file	24.3.0
global.api.ndbdisksize	Default value (30Gi) to be used as in the file	24.3.0
global.ndbapp.ndbdisksize	Default value (20Gi) to be used as in the file	24.3.0

Table 3-18 (Cont.) cnDBTier Customization Parameters

Parameter	Description	Version
mgm.resources.limits.cpu	Default value (1) to be used as in the file	243.0
mgm.resources.limits.memory	Default value (1Gi) to be used as in the file	24.3.0
mgm.resources.requests.cpu	Default value (1) to be used as in the file	24.3.0
mgm.resources.requests.memory	Default value (1Gi) to be used as in the file	24.3.0
ndb.resources.limits.cpu	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • Should be updated to 8 when cnDBTier is planned to be used as extended storage for xDRs • Default value (1) to be used in the file when cnDbTier is not used as extended storage	24.3.0
ndb.resources.limits.memory	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • Should be updated to 128Gi when cnDBTier is planned to be used as extended storage for xDRs • Default value (4Gi) to be used in the file when cnDbTier is not used as extended storage	24.3.0
ndb.resources.requests.cpu	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • Should be updated to 8 when cnDBTier is planned to be used as extended storage for xDRs • Default value (1) to be used in the file when cnDbTier is not used as extended storage	24.3.0
ndb.resources.requests.memory	The default value in the ocnadd_dbtier_custom_values.yaml file to be updated as follows: • Should be updated to 128Gi when cnDBTier is planned to be used as extended storage for xDRs • Default value (4Gi) to be used in the file when cnDbTier is not used as extended storage	24.3.0
api.resources.limits.cpu	Default value (1) to be used as in the file	24.3.0
api.resources.limits.memory	Default value (1Gi) to be used as in the file	24.3.0
api.resources.requests.cpu	Default value (1) to be used as in the file	24.3.0
api.resources.requests.memory	Default value (1Gi) to be used as in the file	24.3.0
api.ndbapp.resources.limits.cpu	Default value (1) to be used as in the file	24.3.0
api.ndbapp.resources.limits.memory	Default value (1Gi) to be used as in the file	24.3.0
api.ndbapp.resources.requests.cpu	Default value (1) to be used as in the file	24.3.0
api.ndbapp.resources.requests.memory	Default value (1Gi) to be used as in the file	24.3.0
db-replicationsvc.dbreplsvcdeployments.enabled	Default value (false) to be used as in the file	24.3.0
db-replicationsvc.resources.limits.cpu	Default value (1) to be used as in the file	24.3.0

Table 3-18 (Cont.) cnDBTier Customization Parameters

Parameter	Description	Version
db-replicationsvc.resources.limits.memory	Default value (2048Mi) to be used as in the file	24.3.0
db-replicationsvc.resources.requests.cpu	Default value (0.6) to be used as in the file	24.3.0
db-replicationsvc.resources.requests.memory	Default value (1024Mi) to be used as in the file	24.3.0
db-monitor-svc.resources.limits.cpu	Default value (200m) to be used as in the file	24.3.0
db-monitor-svc.resources.limits.memory	Default value (500Mi) to be used as in the file	24.3.0
db-monitor-svc.resources.requests.cpu	Default value (200m) to be used as in the file	24.3.0
db-monitorsvc.resources.requests.memory	Default value (500Mi) to be used as in the file	24.3.0

Note

For more information about these parameters, see *Oracle Communications Cloud Native Core cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

4 Upgrading OCNADD

This section provides information on how to upgrade an existing OCNADD deployment. The section describes the upgrade order for source NFs, CNC Console, cnDBTier, and the upgrade impact on the source NFs.

4.1 Migrating OCNADD to New Architecture

This section provides information on how to migrate an existing OCNADD deployment. The section describes the migration order for source NFs, CNC Console, cnDBTier, and the impact on the source NFs.

4.1.1 Migration Overview

The following steps outline the migration process for transitioning an existing OCNADD deployment to the new architecture.

- The migration will follow a Blue-Green deployment approach, where both the old and new architectures coexist.
- A new deployment of the current release will be installed alongside the existing OCNADD deployment, using the same profile to ensure equivalent throughput.
- Once the new deployment installation is verified, configurations will be migrated from the existing deployment to the new one via OCNADD migration job.
- After successful configuration migration, traffic from source NFs will be routed to the new deployment.
- Once the migration is completed and traffic is streaming via the new deployment, normal traffic throughput can be resumed.
- After finalizing the migration, the existing deployment can be scaled down to release the resources.
- After monitoring the new deployment for a sufficient period (typically several days or a week), the existing deployment can be uninstalled. After this point, it is not possible to route traffic via the existing deployment.

> **Note**
>
> - It is assumed that migration is performed at approximately 20% of the currently running traffic rate.
> - The traffic flow between the NFs and the OCNADD Kafka may degrade only when the traffic is being switched from the existing deployment to the new deployment.
> - The traffic flow between OCNADD consumer adapters and third-party consumers may degrade only during the switchover period.
> - Alarms from the source release will not be migrated into the target release during this migration procedure.

4.1.2 Impact on Resource Requirement

The Blue-Green deployment approach requires additional resources during the migration period, as both the old and new deployments will coexist temporarily. However, this approach minimizes the impact on end-to-end traffic flow, with traffic disruption limited to the switchover period. Also, if the migration fails after routing traffic to the new deployment, this approach allows for a quick rollback to the existing deployment with minimal impact. The following resources will be increased during the migration:

- vCPU
- Memory (additional memory will be required if RAM drive storage mode is enabled for Relay Agent or Mediation Kafka cluster. For Relay Agent Kafka, RAM drive storage is enabled by default in the target release)
- Disk Storage PVC for Kraft Controllers deployed in the Relay Agent Kafka cluster
- DB Resources – additional DB resources are required to support the database creation for management group services (`configuration_schema`, `alarm_schema`, and `healthdb_schema`) in the new deployment.

> **Note**
>
> During the migration process, consider the following additional requirements:
>
> - Generate new SSL and TLS certificates for the new deployment and ensure that they are signed using the same CA authority used in the source release. Users can generate a single certificate for each group or a certificate for all services individually.
> - Allocate external IPs for ingress connections (Ingress adapter and Relay Agent Kafka brokers).
> - Update the CNCC console by adding a new instance for the new deployment. If CNCC instance limits are reached, remove the existing deployment instance before adding the new one.

> **Note**
>
> If the user plans to operate at maximum throughput supported in the target release after migration, then the CPU and memory resources required during the migration will be less than the total resources required to support maximum throughput.

4.1.3 Supported Migration Paths

The following table lists the supported migration paths for OCNADD:

Table 4-1 Supported Migration Path

Supported Release	Supported Release
25.2.100	25.2.200
25.1.200	25.2.200

4.1.4 Preparing for migration

Preparing for migration

1. Fetch the images and charts of the target release as described in Pre-Installation Tasks.
2. Keep a backup of the `ocnadd-custom-values.yaml` file and the extracted chart folder `ocnadd` of the source release as a backup before starting the migration procedure.
3. Take the manual backup of the OCNADD before starting the upgrade procedures. See Performing OCNADD Manual Backup for taking a manual backup of the OCNADD.
4. If external access for Kafka brokers is enabled, ensure that you have sufficient IPs in your setup to allocate for the new deployment.
5. While performing the migration, you must align the custom values YAML files of the target release as per the `ocnadd/values.yaml` file of the source release or the older release. Do not enable any new feature during the migration. The parent or sub-charts `values.yaml` must not be changed while performing the upgrade, unless it is explicitly specified in this document. At least the following features must be aligned from source release to target release:
   a. CNLB configurations for OCNADD ingress and egress interface
   b. ACL configurations
   c. Client ACLs (for more details on creating client ACLs, refer to the section *Create Client ACLs* in the *Oracle Communications Network Analytics Data Director User Guide*)
   d. IP Family configurations
   e. IntraTLS and mTLS configurations

   > **Note**
   > - In the target release, the `global.ssl.mTLS` configuration in the `ocnadd-common-custom-values-25.2.200.yaml` file determines whether security is enabled in OCNADD. When set to `true`, security is enabled; otherwise, it is disabled. The default value for this setting is `true`.
   > - If intraTLS and mTLS are set to `false` in the existing deployment, then set `mTLS` to `false` in the target release.
   > - If intraTLS is set to `true` and mTLS is set to `false` in the existing deployment, then set `mTLS` to `true` in the target release.
   > - If intraTLS and mTLS are set to `true` in the existing deployment, then set `mTLS` to `true` in the target release.
6. The database name for Configuration Service, Health Service, and Alarm Service must be modified and should be kept different from the source release in the `ocnadd-common-custom-values-25.2.200.yaml` before the migration:

```
global.cluster.database.configuration_db: configuration_schema          #
--> keep a different name for configuration db than source release
global.cluster.database.alarm_db: alarm_schema                          #
--> keep a different name for alarm db than source release
```

```
global.cluster.database.health_db: healthdb_schema                    #
--> keep a different name for health db than source release
```

For `global.cluster.database.storageadapter_db`, the name should be aligned as per the source release.

7. Ensure to disable the Network Policies before the migration. The network policies can be enabled after the migration. Refer to the section *Network Policy* in the *Oracle Communication Network Analytics Suite Security Guide* for more details.
8. If Druid is enabled in OCNADD for the source release, then Druid configurations must be enabled in the target release, and all required secrets must be created in the management namespace of the target release. For more details on creating secrets, refer to the section *Druid Cluster Integration with OCNADD Site* in the *Oracle Communication Network Analytics Data Director User Guide*.
9. If Export feature is enabled in OCNADD for the source release, then export functionality must be enabled in the target release, and secrets for SFTP credentials must be created in the management namespace of the target release. For more details on creating secrets for SFTP credentials, refer to *Steps to create SFTP credential for SFTP server* in the *Oracle Communication Network Analytics Data Director User Guide*.

4.1.5 Migration Task

4.1.5.1 Choosing the OCNADD Deployment Model

To determine the most suitable deployment model for your use case, refer to the section OCNADD Deployment Models. Prior to initiating the migration process, ensure that the Management Group for both the existing and new deployments is located within the same cluster, as this is a prerequisite for configuration migration. The deployment for the Relay Agent and Mediation Group can be either co-located or distributed across multiple clusters, depending on the selected deployment model.

4.1.5.2 Migration Deployment Considerations (Optional)

Users can deploy Kafka instances and create topic partitions according to the resource profile selected for the target release during the migration. The number of aggregation service instances and adapter instances can be scaled up as needed to accommodate increased throughput. For information on the resource requirements for each profile, refer to the *Oracle Communications Network Analytics Data Director Benchmarking Guide*.

The following example is provided for reference and should not be used as a basis for sizing or configuring any actual deployment:

Let us say the user is deploying the 1500K MPS profile in the target release, and the source NF is SCP.

Profile running in old release: 500K MPS

Profile opted to deploy in new release: 1500K MPS

Example resource profile required for 1500K MPS

```
Relay agent Kafka broker instances: 20
Number of SCP instances: 57
SCP topic partition: 342
```

```
Mediation Kafka broker instances: 20
Number of TCP feed instances: 59
MAIN topic partition: 354
```

When the migration is being performed, resources for the target release should be configured as follows:

```
Relay Agent Kafka replica: 20
SCP topic partition: 342
Number of SCP instances: 20   ## -----> as per 500K MPS profile

Mediation Kafka replica: 20
MAIN topic partition: 354
Number of TCP feed instances: 28     ## -----> as per 500K MPS profile
```

4.1.5.3 Installing and Verifying the OCNADD Deployment

Install the OCNADD package by referring to the section Installing OCNADD Package. While installing OCNADD in the new architecture, it is mandatory to update the worker group name in the custom values of both Relay Agent (`ocnadd-relayagent-custom-values-25.2.200.yaml`) and Mediation (`ocnadd-mediation-custom-values-25.2.200.yaml`) with the worker group namespace name of the source release.

```
global.ocnaddrelayagent.cluster.workergroupName: wg1      # Update with the
namespace name of the worker group in source release
```

Additionally, refer to the section OCNADD UI Configurations Changes for Dashboard Metrics and update the relay agent and mediation groups with the correct worker group name to enable Dashboard metrics in the UI.

Once the installation is completed, verify the installation by referring to the section Verifying OCNADD Installation.

Create Kafka topics and configure topic partitions according to the selected resource profile. For detailed instructions on creating Kafka topics, refer to the Creating OCNADD Kafka Topics section.

4.1.5.4 Migrating Configurations

In this section, migration of all the configurations will be performed from the existing deployment to the newly deployed setup. The configuration will be migrated via the OCNADD migration job. The job will migrate the following OCNADD configurations:

- Standard Feeds
- Ingress Feeds
- Kafka Feeds
- Filter configurations
- Global L3L4 mapping configurations
- OCNADD Metadata configurations
- Correlation and extended storage configuration
- Export configurations

Run the following command in the management group namespace of the target release (25.2.200) to trigger the migration job and start the configuration migration:

```
helm upgrade dd-mgmt -f ocnadd-common-custom-values-25.2.200.yaml -f ocnadd-
management-custom-values-25.2.200-mgmt-group.yaml --namespace
<target_release_management_namespace> --set
global.ocnaddmanagement.migration.enable=true --set
global.ocnaddmanagement.migration.sourceNamespace=<source_release_management_n
amespace> ocnadd
```

For example,

Source release management namespace is `dd-mgmt-old` and target release management namespace is `ocnadd-mgmt`

```
helm upgrade dd-mgmt -f ocnadd-common-custom-values-25.2.200.yaml -f ocnadd-
management-custom-values-25.2.200-mgmt-group.yaml --namespace ocnadd-mgmt --
set global.ocnaddmanagement.migration.enable=true --set
global.ocnaddmanagement.migration.sourceNamespace=dd-mgmt-old ocnadd
```

4.1.5.5 Verify Configuration Migration

Once the migration job is completed, the associated pod for that job will be marked **Completed**. The job will generate a report in the logs of the configurations identified in the source release that were successfully migrated into the target release. To access the report, run the following command

```
kubectl logs -n <target_release_namespace> <migration job podname>
```

Example:

```
kubectl logs -n ocnadd-mgmt ocnaddmigration-hsnr9
```

```
            ========================== EXPORT ============================
            Feature                             | Configurations
            ------------------------------------+------------------------
            ExportConfigurations                | Available

            ====================== dd-old:cluster-1 ======================
            Feature                             | Configurations
            ------------------------------------+------------------------
            IngressAdapterConfigurations        | Available
            Filters                             | Available
            L3L4Mapping                         | Available
            OCNADDMetadata                      | Available
            KafkaFeeds                          | Available
            Configurations                      | Available
            CorrelationConfigurations           | Available

OCL 2025-10-25 10:20:52.251 [main] INFO  c.o.c.c.o.m.s.MigrationService -
            ####### FEATURE: EXPORTCONFIGURATIONS #######
            Config Name                     | Status
            --------------------------------+------------
```

```
CSV-CONFIG                      | SUCCESS

========== Worker Group: dd-old:cluster-1 ===========

####### FEATURE: INGRESSADAPTERCONFIGURATIONS #######
Config Name                     | Status
--------------------------------+------------
ingress-config                  | SUCCESS

####### FEATURE: FILTERS #######
Config Name                     | Status
--------------------------------+------------
Filter Configurations           | SUCCESS

####### FEATURE: L3L4MAPPING #######
Config Name                     | Status
--------------------------------+------------
Gloabl Configuration            | SUCCESS

####### FEATURE: OCNADDMETADATA #######
Config Name                     | Status
--------------------------------+------------
OCNADD MetaData Configuration   | SUCCESS

####### FEATURE: KAFKAFEEDS #######
Config Name                     | Status
--------------------------------+------------
kafkafeed-config                | SUCCESS

####### FEATURE: CONFIGURATIONS #######
Config Name                     | Status
--------------------------------+------------
standard-feed-config            | SUCCESS

####### FEATURE: CORRELATIONCONFIGURATIONS #######
Config Name                     | Status
--------------------------------+------------
kafkafeed-config                | SUCCESS
```

Verify that all the feeds and other configurations are created in the target release. If the job fails due to an error or if all the configurations are not migrated successfully, delete the job manually and re-run the same command provided in the Migrating Configurations section to trigger the execution again.

To get all the jobs in the target release management namespace:

```
kubectl get jobs.batch -n <target_release_management_namespace>
```

To delete the job in the target release management namespace:

```
kubectl delete jobs.batch -n <target_release_management_namespace>
ocnaddmigration
```

4.1.5.6 Configuring OCNADD GUI

Configure the OCNADD UI if not already done by referring to the section Installing OCNADD GUI .

4.1.5.7 Traffic Migration

Once the configurations are migrated successfully, perform the following steps:

- Take the backup of the bootstrap IPs configured in source NFs for the old deployment. This will enable you to restore traffic to the previous deployment in case the migration is unsuccessful.
- Update the bootstrap server in NFs with the Relay Agent Kafka broker IPs/FQDN to migrate traffic to the Relay Agent Kafka cluster in the new deployment.

1. If NFs are deployed in the same cluster and using the FQDN as the Kafka bootstrap to connect to OCNADD, then all the FQDNs must be updated as:
   `*.kafka-broker-headless.<ns>.svc.<domain>`

   where, `<ns>` is the namespace where the Relay Agent Kafka cluster is deployed.

   The asterisk (*) indicates different broker names (`kafka-broker-0`, `kafka-broker-1`, and so on).
2. If NFs are deployed in a different cluster and using IP addresses as the Kafka bootstrap to connect to OCNADD, then the IP addresses in NFs must be updated with the new IP addresses assigned to the Relay Agent Kafka brokers of the target release.
   Only port `9094` on the Kafka broker is supported for establishing connectivity in this access mode.

After the bootstrap server is updated and traffic is redirected to the Relay Agent Kafka cluster, verify the stability of the traffic in the target release by referring to the section Verifying Traffic Migration.

4.1.5.8 Finalizing Migration

Once the migration is complete and traffic is successfully streaming through the Relay Agent Kafka cluster, normal traffic throughput can be restored based on the deployment profile selected by the user. After resuming normal throughput, verify the stability of traffic using the Verifying Traffic Migration section.

> **Note**
>
> Before scaling down the deployments for the Management and Worker groups in the old release, ensure that there is no consumer lag in the Kafka cluster of the old deployment for any worker group.
>
> - If consumer lag is present, the existing deployment must remain active until the lag is fully cleared.
> - If the lag has accumulated due to a connection failure between feeds and a third-party application, ensure that the connectivity issue is resolved and **all** lag is cleared before proceeding.

1. Scaling down all worker group resources (Source Release)

a. **Scale down Deployments**
Run the following command for **every deployment** in the source worker group namespace:

```
kubectl scale deploy <deployment_name> -n
<source_release_worker_group_namespace> --replicas 0
```

b. **Scale down StatefulSets**
Run the following command for **every StatefulSet (sts)** in the source worker group namespace:

```
kubectl scale sts <sts_name> -n <source_release_worker_group_namespace>
--replicas 0
```

2. Scaling down management group resources (Source Release)
Run the following command for **every deployment** in the source management group namespace:

```
kubectl scale deploy <deployment_name> -n
<source_release_management_group_namespace> --replicas 0
```

4.1.5.9 Verifying Traffic Migration

To confirm that traffic is running stably after migration, verify the following key points:

1. **Pod stability:** Ensure that no pods are restarting or entering crash loops after beginning to receive traffic.
2. **Throughput metrics:** Monitor both Ingress and Egress throughput to confirm that the expected Messages Per Second (MPS) rate has been achieved.
3. **Resource utilization:** Check Pod CPU and Memory consumption to ensure that utilization remains within acceptable limits and no resource bottlenecks exist.
4. **Kafka consumer performance:** Verify Kafka **Consumer Lag** to ensure that consumers are processing messages at the required rate.
5. **End-to-end latency:** Measure OCNADD end-to-end latency to assess overall traffic processing performance.

For detailed steps on collecting the necessary data, refer to the section **"Troubleshooting Traffic Stability in OCNADD"** in the *Oracle Communications Network Analytics Data Director Troubleshooting Guide*.

If any **critical anomalies** are observed in the metrics or command outputs that significantly impact traffic in the new architecture, the user may redirect traffic back to the existing OCNADD deployment to avoid service disruption. To route traffic back to the old deployment, follow the steps below:

1. **Reconfigure the bootstrap IPs** in the source NF with the bootstrap IPs of the old OCNADD deployment.
2. **Verify traffic** on the old deployment using throughput metrics.

During this time, the user can troubleshoot issues in the new deployment. For guidance on resolving post-migration issues, refer to the *Oracle Communications Network Analytics Data Director Troubleshooting Guide*.

4.1.6 Post Migration Task

> **⚠ Caution**
>
> Performing this task is irreversible and will prevent you from routing traffic back to the existing deployment. Proceed only when you are completely satisfied with the new deployment and have no intention of reverting to the previous version.
>
> After this procedure is performed, if the user needs to revert to the previous release, then the user will have to perform a *Fault Recovery* on the older release.

After the migration is completed, monitor the new deployment for a sufficient period of time. Most users typically monitor the deployment for several days to a week. Once the user has monitored the new deployment for a sufficient amount of time, the user can uninstall the existing deployment using the following steps:

1. **Uninstall the worker groups one after another using the following command:**

   ```
   helm uninstall <worker-group-release-name> --namespace <worker-group-namespace>
   ```

   Example:

   ```
   helm uninstall ocnadd-wg1 -namespace dd-worker-group1
   ```

2. **Clean up Kafka Configuration for all the worker groups.**
   a. To list the secrets in the namespace, run:

      ```
      kubectl get secrets -n <worker-group-namespace>
      ```

   b. To delete all the secrets related to Kafka, run:

      ```
      kubectl delete secret --all -n <worker-group-namespace>
      ```

   c. To delete the configmap used for Kafka, run:

      ```
      kubectl delete configmap --all -n <worker-group-namespace>
      ```

   d. To delete PVCs used for Kafka:
      i. Run the following command to list the PVCs used in the namespace:

         ```
         kubectl get pvc -n <worker-group-namespace>
         ```

      ii. Run the following command to delete the PVCs used by the brokers and zookeepers:

         ```
         kubectl delete pvc --all -n <worker-group-namespace>
         ```

3. **Delete all the worker group namespaces using the below command (This step is only needed if there is more than one worker group):**

   ```
   kubectl delete namespace <worker-group-namespace>
   ```

4. **Uninstall the management group using the following command:**

   ```
   helm uninstall <management-release-name> --namespace <management-group-namespace>
   ```

   Example:

   ```
   helm uninstall ocnadd-mgmt --namespace dd-mgmt-group
   ```

5. **Clean up Database**
   a. Log in to the MySQL client on SQL Node with the OCNADD user and password:

      ```
      mysql -h <IP_address of SQL Node> -u <ocnadduser> -p
      ```

   b. To clean up the configuration, alarm, and health database, run:

      ```
      DROP DATABASE <dbname>;
      ```

   c. To remove MySQL users while uninstalling OCNADD, run:

      ```
      SELECT user FROM mysql.user;
      DROP USER 'ocnaddappuser@'%';
      ```

4.2 Post Upgrade Task

> **Note**
>
> This step is required only when the OCCM is used to manage the certificates in source and target releases and the user wants to update the Loadbalancer IPs for a service in the target release. For step-by-step details, refer to the section Adding or Updating Load Balancer IPs in SAN When OCCM is Used.

4.2.1 Druid Cluster Integration with OCNADD Site

Druid Cluster Integration with OCNADD Site

> **Note**
>
> In the previous release(s) where the extended storage was available only using the cnDBTier database, the migration from cnDBTier-based extended storage to Druid-based extended storage is not supported. In case the user wants to move to the Druid-based extended storage from cnDBTier-based extended storage, the user must remove the correlation configurations, export and trace configurations before integrating the Druid-based extended storage. After the Druid storage has been integrated with the OCNADD site, the user can create the correlation, export and trace configuration again.

This feature is introduced as part of extended storage in the Data Director. To enable it, refer to the *Druid Cluster Integration with OCNADD* section in the *Oracle Communications Network Analytics Data Director User Guide*. The feature is recommended to be enabled after the release upgrade is completed. The extended storage using the cnDBTier database is available by default if this Druid cluster integration is not enabled.

4.2.2 vCollector Integration for Diameter Feed

In this release, the integration with vCollector is provided. The vCollector acquires the Diameter traffic from vDSR using port mirroring. vCollector is deployed as a virtual machine outside the OCNADD cluster and provides the acquired Diameter traffic to Data Director over the Kafka interface. The vCollector is configured and managed by the Data Director OAM services. This feature is introduced as part of Diameter feed capabilities in the Data Director. To enable the integration with vCollector, refer to the *vCollector Integration with Data Director* section in the *Oracle Communications Network Analytics Data Director Diameter User Guide*. The feature is recommended to be enabled after the release installation is completed.

5

Rolling Back OCNADD

The upgrade for OCNADD is managed as a migration; therefore, a Helm-based rollback is not feasible for the current release. In the event of a failed migration, a rollback to the old deployment can be achieved by routing the traffic from source NFs back to the old deployment, provided that the Post-Migration Task has not yet been performed and the old deployment is still present. If the Post-Migration Task has been completed and the previous deployment is not available, rollback can be performed by executing fault recovery on the older release.

6
Uninstalling OCNADD

This chapter provides information on how to uninstall Oracle Communications Network Analytics Data Director (OCNADD).

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

> **Note**
>
> `kubectl` commands might vary based on the platform deployment. Replace `kubectl` with Kubernetes environment-specific command line tool to configure kubernetes resources through kube-api server. The instructions provided in this document are as per the Oracle Communications Cloud Native Core, Cloud Native Environment (CNE) version of kube-api server.

> **Caution**
>
> - While deleting any OCNADD resources make sure to provide the corresponding namespace used in the deployment.
> - Based on requirement, make sure to retain the OCNADD backup before the uninstallation procedure. For more information, see Performing OCNADD Backup Procedures.
> - Ensure any configured datafeeds are deleted using the OCNADD GUI prior to performing the OCNADD uninstallation steps. For deletion of the datafeeds, refer to *Oracle Communications Network Analytics Data Director User Guide*.
> - The command `kubectl delete all` deletes all the Kubernetes objects in the specified namespace. In case you have created the RBAC resources and service accounts before the Helm installation in the same namespace, and these resources are required, then do not delete them.
> - The command `kubectl delete namespace` removes all the resources or objects created in the namespace. Therefore, ensure that you run the command only when you want to delete the namespace completely.

6.1 Uninstalling Worker Group

To uninstall a worker group, all the associated sub-groups (relay agent and mediation) should be uninstalled. The steps below are listed only once, but they should be repeated for all relay agent and mediation groups that should be uninstalled. All mediation groups should be uninstalled one after another, and then the relay agent group should be uninstalled. Replace the `<ocnadd-group-namespace>` with the relay agent or mediation group namespace.

1. Run the following command to uninstall the OCNADD group:

   ```
   helm uninstall <ocnadd-group-release-name> --namespace <ocnadd-group-namespace>
   ```

   For example:

   ```
   helm uninstall dd-med --namespace ocnadd-med
   helm uninstall dd-ra --namespace ocnadd-relay
   ```

2. Clean up Kafka Configuration for all mediation and relay agent groups.
   To clean up the Kafka configuration, perform the following steps:
   a. To list the secrets in the namespace, run:

      ```
      kubectl get secrets -n <ocnadd-group-namespace>
      ```

   b. To delete all the secrets related to Kafka, run:

      ```
      kubectl delete secret --all -n <ocnadd-group-namespace>
      ```

   c. To delete the configmap used for Kafka, run:

      ```
      kubectl delete configmap --all -n <ocnadd-group-namespace>
      ```

   d. To delete PVCs used for Kafka:
      i. Run the following command to list the PVCs used in the namespace:

         ```
         kubectl get pvc -n <ocnadd-group-namespace>
         ```

      ii. Run the following command to delete the PVCs used by the brokers and Kraft controllers:

         ```
         kubectl delete pvc --all -n <ocnadd-group-namespace>
         ```

3. Run the following command to delete all the objects:
   a. To delete all the Kubernetes objects:

      ```
      kubectl delete all --all -n <ocnadd-group-namespace>
      ```

   b. Run the following command to delete specific resources:

      ```
      kubectl delete <resource-type> <resource-name> -n <ocnadd-group-namespace>
      ```

4. Delete all the relay agent and mediation group namespaces using the command:

   ```
   kubectl delete namespace <ocnadd-group-namespace>
   ```

6.2 Uninstalling Management Group

1. Uninstall the management group using the following command:

   ```
   helm uninstall <management-release-name> --namespace <management-group-namespace>
   ```

   For example:

   ```
   helm uninstall dd-mgmt --namespace ocnadd-mgmt
   ```

2. Check the management group namespace:

   ```
   kubectl get all -n <management-group-namespace>
   ```

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

   If the command output displays OCNADD resources or objects, then perform the following procedure to delete all the objects:

   a. To delete all the Kubernetes objects:

      ```
      kubectl delete all --all -n <mgmt-group-namespace>
      ```

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

      ```
      kubectl delete <resource-type> <resource-name> -n <management-group-namespace>
      ```

   c. Run the following command to delete the management group namespace:

      ```
      kubectl delete namespace <management-group-namespace>
      ```

      For example:

      ```
      kubectl delete namespace ocnadd-mgmt
      ```

3. Clean up the Database.
   To clean up the database, perform the following steps:

   a. Log in to the MySQL client on the SQL Node with the `ocnadduser` and password:

      ```
      mysql -h <IP_address of SQL Node> -u <ocnadduser> -p
      ```

   b. To clean up the configuration, alarm, and health database, run the following command and pass the database names:

      ```
      mysql> drop database <dbname>;
      ```

c. To remove MySQL users while uninstalling OCNADD, run the following commands:

```
SELECT user FROM mysql.user;
DROP USER 'ocnaddappuser@'%';
```

6.3 Verifying Uninstallation

To verify the OCNADD uninstallation, run the following command:

Check if any management, relay agent, or mediation group namespaces exist:

```
kubectl get namespaces
```

The output should not list any namespaces for OCNADD groups that are uninstalled.

7

Migrating to OCCM Managed Certificates

> **⚠ Caution**
>
> - It is expected that there will be downtime when the services are migrated to use the new certificates generated by the OCCM. The amount of downtime will depend on the method of migration performed as described below.
> - This procedure is applicable when certificates are being migrated within the same release.
> - Migration supported only for current release version

This section provides information on how to migrate the certificates initially created by following the section "Configuring SSL or TLS Certificates" during OCNADD installation.

The below steps can be followed to use certificates created by OCCM:

7.1 Upgrading the Helm Charts

> **⚠ Caution**
>
> No configuration or existing data will be lost. The expected downtime will be equal to the time taken to upgrade the relay agent group, the mediation group, the consumer adapter, and correlation, plus the time required for the Kafka broker and KRaft controller to stabilize. The Kafka broker and KRaft controller must be stabilized for all available groups.

To manually create certificates for OCNADD, follow these steps:

1. Follow the steps to create secrets for OCCM for each management and worker group (relay agent and mediation) namespace as specified in the "OCCM Pre-requisites for Installing OCNADD" section in the *Oracle Communications Network Analytics Suite Security Guide*.
2. Enable the OCCM based certificate management in the Management and Worker group (relay agent and mediation) custom-values. For descriptions of the Helm parameters required for enabling OCCM, see Helm Parameter Configuration for OCCM.
3. Upgrade the Management group helm chart:

   ```
   helm upgrade <management-release-name> -f ocnadd-common-custom-values.yaml
   -f ocnadd-management-custom-values.yaml --namespace <management-group-
   namespace> <helm_chart>
   ```

For example:

```
helm upgrade dd-mgmt -f ocnadd-common-custom-values.yaml -f ocnadd-
management-custom-values.yaml --namespace dd-mgmt-group ocnadd_mgmt
```

4. Upgrade the Relay Agent group helm chart:

```
helm upgrade <relayagent-release-name> -f ocnadd-common-custom-values.yaml
-f ocnadd-relayagent-custom-values.yaml --namespace <relayagent-group-
namespace> <helm_chart>
```

For example:

```
helm upgrade dd-rea -f ocnadd-common-custom-values.yaml -f ocnadd-
relayagent-custom-values.yaml --namespace ocnadd-relay ocnadd
```

5. Upgrade the Mediation group helm chart:

```
helm upgrade <mediation-release-name> -f ocnadd-common-custom-values.yaml -
f ocnadd-mediation-custom-values.yaml --namespace <mediation-group-
namespace> <helm_chart> --set
global.ocnaddmediation.env.admin.OCNADD_INGRESS_ADAPTER_UPGRADE_ENABLE=true
,global.ocnaddmediation.env.admin.OCNADD_ADAPTER_UPGRADE_ENABLE=true,global
.ocnaddmediation.env.admin.OCNADD_CORR_UPGRADE_ENABLE=true,global.ocnaddmed
iation.env.admin.OCNADD_STORAGE_ADAPTER_UPGRADE_ENABLE=true
```

For example:

```
helm upgrade dd-med -f ocnadd-common-custom-values.yaml -f ocnadd-
mediation-custom-values.yaml --namespace ocnadd-med ocnadd --set
global.ocnaddmediation.env.admin.OCNADD_INGRESS_ADAPTER_UPGRADE_ENABLE=true
,global.ocnaddmediation.env.admin.OCNADD_ADAPTER_UPGRADE_ENABLE=true,global
.ocnaddmediation.env.admin.OCNADD_CORR_UPGRADE_ENABLE=true,global.ocnaddmed
iation.env.admin.OCNADD_STORAGE_ADAPTER_UPGRADE_ENABLE=true
```

6. If multiple mediation groups are present repeat steps 5 for each mediation group

8
Fault Recovery

This chapter provides information about fault recovery for OCNADD deployment.

8.1 Overview

This section describes procedures to perform the backup and restore for the Oracle Communications Network Analytics Data Director (OCNADD) deployment. The backup and restore procedures will be used in the fault recovery of OCNADD. The OCNADD operators can take only the OCNADD-instance-specific database and required OCNADD Kafka metadata backup and restore them either on the same or a different Kubernetes cluster.

The backup and restore procedures are helpful in the following scenarios:

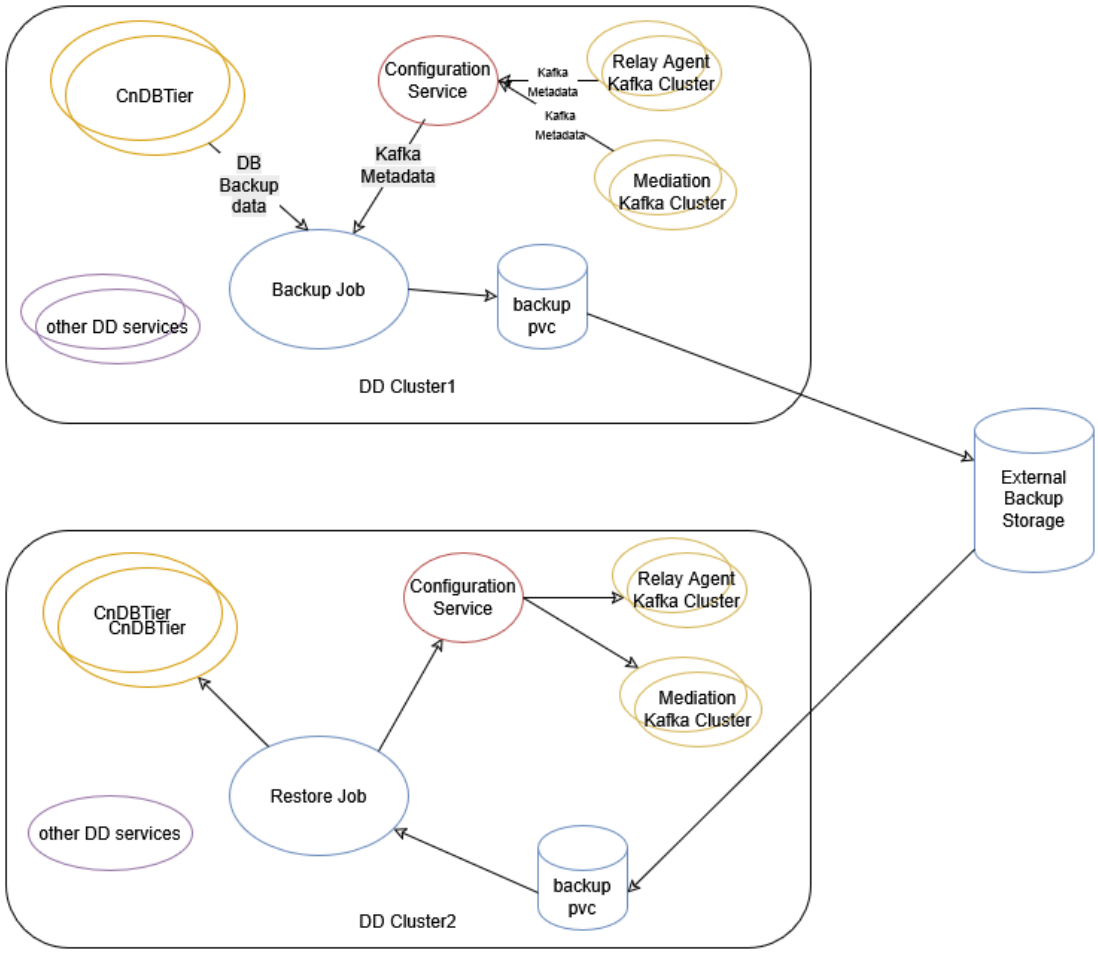
- OCNADD fault recovery
- OCNADD cluster migration
- OCNADD setup replication from production to development or staging
- OCNADD cluster upgrade to a new CNE version or K8s version

The OCNADD backup contains the following data:

- OCNADD database(s) backup
- OCNADD Kafka metadata backup, including the topics and partitions information

> **Note**
>
> If the deployed Helm charts and the customized custom values for the management, relay agent, and mediation groups for the current deployment are stored in the Helm or artifact repository, then the Helm chart and custom values backup is not required. To successfully execute backup and restore operations for Kafka metadata, it is essential that the configuration service is operational and running. The absence of a running configuration service prevents Kafka metadata backup and restore from being performed.

Figure 8-1 OCNADD Backup and Restore



OCNADD Database(s) Backup

The OCNADD database consists of the following:

- Configuration data: This data is exclusive to the given OCNADD instance. Therefore, an exclusive logical database is created and used by an OCNADD instance to store its configuration data and operator-driven configuration. Operators can configure the OCNADD-instance-specific configurations using the Configuration UI service through the Cloud Native Configuration Console.
- Health monitoring data: This data is also exclusive to the given OCNADD instance. Therefore, an exclusive logical database is created and used by an OCNADD Health Monitoring service instance to store the health profile of various other services.

The database backup job uses the `mysqldump` utility.

Scheduled regular backups help in:

- Restoring the stable version of the data directory databases
- Minimizing significant loss of data due to upgrade or rollback failure
- Minimizing loss of data due to system failure
- Minimizing loss of data due to data corruption or deletion due to external input

- Migration of the database information from one site to another site

OCNADD Kafka Metadata Backup

The OCNADD Kafka metadata backup contains the following information:

- Created topics information
- Created partitions per topic information

8.1.1 Fault Recovery Impact Areas

The following table shares information about impact of OCNADD fault recovery scenarios:

Table 8-1 OCNADD Fault Recovery Scenarios Impact Information

Scenario	Requires Fault Recovery or Reinstallation of CNE?	Requires Fault Recovery or Reinstallation of cnDBTier?	Requires Fault Recovery or Reinstallation of Data Director?
Scenario 1: Deployment Failure Recovering OCNADD when its deployment is corrupted	No	No	Yes
Scenario 2: cnDBTier Corruption	No	Yes	No However, it requires to restore the databases from backup and Helm upgrade of the same OCNADD version to update the OCNADD configuration. For example, change in cnDBTier service information, such as cnDB endpoints, DB credentials, and so on.
Scenario 3: Database Corruption Recovering from corrupted OCNADD configuration database	No	No	No However, it requires to restore the databases from old backup.
Scenario 4: Site Failure Complete site failure due to infrastructure failure, for example, hardware, CNE, and so on.	Yes	Yes	Yes

8.1.2 Prerequisites

Before you run any fault recovery procedure, ensure that the following prerequisites are met:

- cnDBTier must be in a healthy state and available on a new or newly installed site where the restore needs to be performed.

- Do not change `DB Secret` or cnDBTier MySQL FQDN or IP or PORT configurations during backup and restore.
- Automatic backup should be enabled for OCNADD.
- Docker images used during the last installation or upgrade must be retained in the external data storage or repository.
- The management, relay agent, and mediation group custom values file used at the time of the OCNADD deployment must be retained. If the custom values files are not retained, they must be recreated manually. This task increases the overall fault recovery time.

8.2 Backup and Restore Flow

Important

- It is recommended to keep the backup storage in the external storage that can be shared between different clusters. This is required, so that in an event of a fault, the backup is accessible on the other clusters. The backup job should create a PV or PVC from the external storage provided for the backup.
- In case the external storage is not made available for the backup storage, the customer should take care to copy the backups from the associated backup PV in the cluster to the external storage. The security and connectivity to the external storage should be managed by the customer. To copy the backup from the backup PV to the external server, follow Verifying OCNADD Backup.
- The restore job should have access to the external storage so that the backup from the external storage can be used for the restoration of the OCNADD services. In case the external storage is not available, the backup should be copied from the external storage to the backup PV in the new cluster. For information on the procedure, see Verifying OCNADD Backup.
- In case of two site redundancy feature is enabled then respective site backup should be used to restore the site during failure recovery.

> **Note**
>
> At a time, only one of the three backup jobs (`ocnaddmanualbackup`, `ocnaddverify`, or `ocnaddrestore`) can be running. If any existing backup job is running, that job needs to be deleted to spawn the new job.
>
> ```
> kubectl delete job.batch/<ocnadd*> -n <namespace>
> ```
>
> where,
>
> - `namespace` = Namespace of OCNADD deployment
> - `ocnadd*` = Running jobs in the namespace (`ocnaddmanualbackup`, `ocnaddverify`, or `ocnaddrestore`)
>
> Example:
>
> ```
> kubectl delete job.batch/ocnaddverify -n ocnadd-deploy
> ```

Backup

1. The OCNADD backup is managed using the backup job created at the time of installation. The backup job runs as a cron job and takes the daily backup of the following:
   - OCNADD databases for configuration, alarms, and health monitoring
   - OCNADD Kafka metadata including topics and partitions, which are previously created
2. The automated backup job spawns as a container and takes the backup at the scheduled time. The backup file `OCNADD_Backup_DD-MM-YYYY_hh-mm-ss.tar.bz2` is created and stored in the PV mounted on the path `/work-dir/backup` by the backup container.
3. On-demand backup can also be created by creating the backup container. For more information, see Performing OCNADD Manual Backup.
4. The backup can be stored on external storage.

Restore

1. The OCNADD restore job must have access to the backups from the backup PV/PVC.
2. The restore uses the latest backup file available in the backup storage if the BACKUP_FILE argument is not given.
3. The restore job performs the restore in the following order:
   a. Restore the OCNADD database(s) on the cnDBTier.
   b. Restore the Kafka metadata.

8.3 OCNADD Backup

The OCNADD backup is of two types:

- Automated backup
- Manual backup

Automated Backup

- This is managed by the automated K8s job configured during the installation of the OCNADD. For more information, see Updating the OCNADD Backup Cronjob step.
- It is a scheduled job and runs daily at the configured time to collect the OCNADD backup and creates the backup file `OCNADD_Backup_DD-MM-YYYY_hh-mm-ss.tar.bz2`.

Manual Backup

- This is managed by an on-demand job.
- A new K8s job will be created on executing the Performing OCNADD Manual Backup procedure.
- The job completes after taking the backup. Follow Verifying OCNADD Backup procedure to verify the generated backup.

8.4 Performing OCNADD Backup Procedures

8.4.1 Performing OCNADD Manual Backup

Perform the following steps to take the manual backup:

1. Go to the `custom_templates` folder in the extracted OCNADD release package and update the `ocnadd_manualBackup.yaml` file, or the `ocnadd_manualBackup_occm.yaml` file if OCCM is used, with the following information:
   a. Value for `BACKUP_DATABASES` can be set to `ALL` (that is, `configuration_schema` and `healthdb_schema`), or the individual DB names can also be passed. By default, the value is `ALL`.
   b. Value of `BACKUP_ARG` can be set to `ALL`, `DB`, or `KAFKA`. By default, the value is `ALL`.
   c. Update other values as follows:

```
apiVersion: batch/v1
kind: Job
metadata:
   name: ocnaddmanualbackup
   namespace: ocnadd-mgmt          #---> update the management
namespace
------------
spec:
       serviceAccountName: ocnadd-mgmt-sa-ocnadd  #---> update the
service account name. Format:<serviceAccount>-sa-ocnadd
------------
       containers:
       - name: ocnaddmanualbackup
       image: <repo-path>/ocdd.repo/ocnaddbackuprestore:2.0.9 #--->
update repository path
------------
       initContainers:
       - name: ocnaddinitcontainer
       image: <repo-path>/utils.repo/jdk21-openssl:1.0.9 #---> update
repository path
       env:
```

```
- name: BACKUP_DATABASES
  value: ALL
- name: BACKUP_ARG
  value: ALL
```

If **(1) Single Certs for each component** is selected as the certificate generation process, run the below commands:

i. When `generate_certs.sh` is used:

```
sed -i 's/ocnaddbackuprestore-secret/ocnadd-secret/g'
ocnadd_manualBackup.yaml
sed -i 's/ocnaddbackuprestore-servercert/ocnadd-servercert/g'
ocnadd_manualBackup.yaml
sed -i 's/ocnaddbackuprestore-serverprivatekey/ocnadd-
serverprivatekey/g' ocnadd_manualBackup.yaml
```

ii. When OCCM is used:

```
sed -i 's/ocnaddbackuprestore-secret/ocnadd-secret/g'
ocnadd_manualBackup_occm.yaml
sed -i 's/ocnaddbackuprestore-servercert/ocnadd-servercert/g'
ocnadd_manualBackup_occm.yaml
sed -i 's/ocnaddbackuprestore-serverprivatekey/ocnadd-
serverprivatekey/g' ocnadd_manualBackup.yaml
```

2. Run the following command to run the job:

```
kubectl create -f ocnadd_manualBackup.yaml
```

OR, if OCCM is used:

```
kubectl create -f ocnadd_manualBackup_occm.yaml
```

8.4.2 Verifying OCNADD Backup

> ⚠ **Caution**
>
> The connectivity between the external storage through either PV/PVC or network connectivity must be ensured.

To verify the backup, perform the following steps:

1. Go to the `custom_templates` folder in the extracted OCNADD release package and update the `ocnadd_verify_backup.yaml` file, or the `ocnadd_verify_backup_occm.yaml` file if OCCM is used, with the following information:
   a. Sleep time is configurable, update it if required (the default value is set to 10m).
   b. Update other values as follows:

```
apiVersion: batch/v1
kind: Job
```

```
metadata:
   name: ocnaddverify
   namespace: ocnadd-mgmt                                          #--->
update the management namespace
------------
spec:
      serviceAccountName: ocnadd-mgmt-sa-ocnadd       -
#---> update the service account name. Format:<serviceAccount>-sa-
ocnadd
------------
      containers:
      - name: ocnaddverify
      image: <repo-path>/ocdd.repo/ocnaddbackuprestore:2.0.9       #---
> update repository path
------------
      initContainers:
      - name: ocnaddinitcontainer
      image: <repo-path>/utils.repo/jdk21-openssl:1.0.9            #---
> update repository path
```

If (1) Single Certs for each component is selected as the certificate generation process, run the below commands:

a. When `generate_certs.sh` is used

```
sed -i 's/ocnaddbackuprestore-secret/ocnadd-secret/g'
ocnadd_verify_backup.yaml
sed -i 's/ocnaddbackuprestore-servercert/ocnadd-servercert/g'
ocnadd_verify_backup.yaml
sed -i 's/ocnaddbackuprestore-serverprivatekey/ocnadd-
serverprivatekey/g' ocnadd_manualBackup.yaml
```

b. When OCCM is used

```
sed -i 's/ocnaddbackuprestore-secret/ocnadd-secret/g'
ocnadd_verify_backup_occm.yaml
sed -i 's/ocnaddbackuprestore-servercert/ocnadd-servercert/g'
ocnadd_verify_backup_occm.yaml
sed -i 's/ocnaddbackuprestore-serverprivatekey/ocnadd-
serverprivatekey/g' ocnadd_verify_backup_occm.yaml
```

2. Run the below command to create the job:

```
kubectl create -f ocnadd_verify_backup.yaml
```

Or, use the following command if OCCM is used:

```
kubectl create -f ocnadd_verify_backup_occm.yaml
```

3. If the external storage is used as PV/PVC, then enter the ocnaddverify-xxxx container using the following commands:

a. `kubectl exec -it <ocnaddverify-xxxx> -n <ocnadd namespace> -- bash`

b. Change the directory to `/work-dir/backup` and inside the latest backup file `OCNADD_BACKUP_DD-MM-YYYY_hh-mm-ss.tar.bz2`, verify the DB backup and Kafka metadata backup files.

8.4.3 Retrieving the OCNADD Backup Files

1. Run the Verifying OCNADD Backup procedure to spawn the `ocnaddverify-xxxx`.
2. Go to the running `ocnaddverify` pod to identify and retrieve the desired backup folder using the following commands:
   a. Run the following command to access the pod:

      ```
      kubectl exec -it <ocnaddverify-xxxx> -n <ocandd-namespace> -- bash
      ```

      where,

      `<ocnadd-namespace>` is the namespace where the ocnadd management group services are running.

      `<ocnaddverify-xxxx>` is the backup verification pod in the same namespace.

   b. Change the directory to `/work-dir/backup` and identify the backup file "`OCNADD_BACKUP_DD-MM-YYYY_hh-mm-ss.tar.bz2`".

   c. Exit the `ocnaddverify` pod.
3. Copy the backup file from the pod to the local bastion server by copying the file `OCNADD_BACKUP_DD-MM-YYYY_hh-mm-ss.tar.bz2`, and run the following command:

   ```
   kubectl cp -n <ocnadd-namespace> <ocnaddverify-xxxx>:/work-dir/backup/
   <OCNADD_Backup_DD-MM-YYYY_hh-mm-ss.tar.bz2> <OCNADD_Backup_DD-MM-YYYY_hh-
   mm-ss.tar.bz2>
   ```

   where,

   `<ocnadd-namespace>` is the namespace where the ocnadd management group services are running.

   `<ocnaddverify-xxxx>` is the backup verification pod in the same namespace.

   For example:

   ```
   kubectl cp -n ocnadd-mgmt ocnaddverify-drwzq:/work-dir/backup/
   OCNADD_BACKUP_10-05-2023_08-00-05.tar.bz2
   OCNADD_BACKUP_10-05-2023_08-00-05.tar.bz2
   ```

8.4.4 Copying and Restoring the OCNADD backup

1. Retrieve the OCNADD backup file.
2. Perform the Verifying OCNADD Backup procedure to spawn the `ocnaddverify-xxxx`.

3. Copy the backup file from the local bastion server to the running ocnaddverify pod, run the following command:

   ```
   kubectl cp <OCNADD_Backup_DD-MM-YYYY_hh-mm-ss.tar.bz2> <ocnaddverify-xxxx>:/work-dir/backup/<OCNADD_Backup_DD-MM-YYYY_hh-mm-ss.tar.bz2> -n <management-group-namespace>
   ```

   For example:

   ```
   kubectl cp OCNADD_BACKUP_10-05-2023_08-00-05.tar.bz2 ocnaddverify-mrdxn:/work-dir/backup/OCNADD_BACKUP_10-05-2023_08-00-05.tar.bz2 -n ocnadd-mgmt
   ```

4. Go to `ocnaddverify` pod and path, `/workdir/backup/OCNADD_BACKUP_DD-MM-YYYY_hh-mm-ss.tar.bz2` to verify if the backup has been copied.
5. Restore OCNADD using the procedure defined in Creating OCNADD Restore Job.

8.5 Fault Recovery Scenarios

This chapter describes the fault recovery procedures for different recovery scenarios.

8.5.1 Scenario 1: Deployment Failure

This section describes how to recover OCNADD when the OCNADD deployment corrupts.

For more information, see Restoring OCNADD.

8.5.2 Scenario 2: cnDBTier Corruption

This section describes how to recover the cnDBTier corruption. For more information, see *Oracle Communications Cloud Native Core cnDBTier Installation, Upgrade, and Fault Recovery Guide*. After the cnDBTier recovery, restore the OCNADD database from the previous backup.

To restore the OCNADD database, execute the procedure Creating OCNADD Restore Job by setting BACKUP_ARG to DB.

8.5.3 Scenario 3: Database Corruption

This section describes how to recover from the corrupted OCNADD database.

Perform the following steps to recover the OCNADD configuration database (DB) from the corrupted database:

1. Retain the working ocnadd backup by following Retrieving the OCNADD Backup Files procedure.
2. Drop the existing Databases by accessing the MySql DB.
3. Perform the Copying and Restoring the OCNADD backup procedure to restore the backup.

8.5.4 Scenario 4: Site Failure

This section describes how to perform fault recovery when the OCNADD site has software failure.

Perform the following steps in case of a complete site failure:

1. Run the Cloud Native Environment (CNE) installation procedure to install a new Kubernetes cluster. For more information, see *Oracle Communications Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.
2. Run the cnDBTier installation procedure. For more information, see *Oracle Communications Cloud Native Core cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
3. For cnDBTier fault recovery, take a data backup from an older site and restore it to a new site. For more information about cnDBTier backup, see "Create On-demand Database Backup" and to restore the database to a new site, see "Restore DB with Backup" in *Oracle Communications Cloud Native Core cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
4. Restore OCNADD. For more information, see Restoring OCNADD.

8.6 Restoring OCNADD

Perform this procedure to restore OCNADD when a fault event has occurred or deployment is corrupted.

> **Note**
>
> This procedure expects the OCNADD backup folder is retained.

1. Get the retained backup file "`OCNADD_BACKUP_DD-MM-YYYY_hh-mm-ss.tar.bz2`".
2. Get the Helm charts that was used in the earlier deployment.
3. Run the following command to uninstall the corrupted OCNADD deployment: Management Group or Worker Group (Relay Agent or Mediation):

   ```
   helm uninstall <release_name> --namespace <namespace>
   ```

   Where,

   `<release_name>` is the release name of the ocnadd deployment which is being uninstalled.

   `<namespace>` is the namespace of OCNADD deployment which is being uninstalled.

   For example: To uninstall the Management Group

   ```
   helm  uninstall ocnadd-mgmt --namespace dd-mgmt-group
   ```

4. Install the Management Group or Worker Group (Relay Agent or Mediation) that was corrupted and uninstalled in the previous step using the helm charts that were used in the earlier deployment. For the installation procedure see, Installing OCNADD.
5. To verify whether OCNADD installation is complete, see Verifying OCNADD Installation.
6. Follow procedure Copying and Restoring the OCNADD backup

8.7 Creating OCNADD Restore Job

Follow the below steps to create and run OCNADD restore job:

1. Restore the OCNADD database by following below steps:
   a. Go to the `custom_templates` folder inside the extracted `ocnadd-release` package and update the `ocnadd_restore.yaml` or the `ocnadd_restore_occm.yaml` file if OCCM is used, with the following information:
      i. The value of BACKUP_ARG can be set to DB, KAFKA, and ALL. By default, the value is 'ALL'.
      ii. The value of BACKUP_FILE can be set to folder name which needs to be restored, if not mentioned the latest backup will be used.
      iii. Update other values as below:

```
apiVersion: batch/v1
kind: Job
metadata:
   name: ocnaddrestore
   namespace: ocnadd-mgmt     #---> update the management namespace
------------
spec:
      serviceAccountName: ocnadd-mgmt-sa-ocnadd #---> update the
service account name. Format:<serviceAccount>-sa-ocnadd
------------
      containers:
      - name: ocnaddrestore
      image: <repo-path>/ocdd.repo/ocnaddbackuprestore:2.0.9 #--->
update repository path
------------
      initContainers:
      - name: ocnaddinitcontainer
      image: <repo-path>/utils.repo/jdk21-openssl:1.0.9 #--->
update repository path
      env:
          - name: BACKUP_ARG
            value: ALL
          - name: BACKUP_FILE
            value: "" #---> update the backup file name which needs
to be restored, if not mentioned the latest backup will be used for
example "OCNADD_Backup_DD-MM-YYYY_hh-mm-ss.tar.bz2"
```

      If (1) Single Certs for each component is selected as the certificate generation process, run the below commands:

      i. When `generate_certs.sh` is used

```
sed -i 's/ocnaddbackuprestore-secret/ocnadd-secret/g'
ocnadd_restore.yaml
sed -i 's/ocnaddbackuprestore-servercert/ocnadd-servercert/g'
ocnadd_restore.yaml
sed -i 's/ocnaddbackuprestore-serverprivatekey/ocnadd-
serverprivatekey/g' ocnadd_restore.yaml
```

ii. When OCCM is used

```
sed -i 's/ocnaddbackuprestore-secret/ocnadd-secret/g'
ocnadd_restore_occm.yaml
sed -i 's/ocnaddbackuprestore-servercert/ocnadd-servercert/g'
ocnadd_restore_occm.yaml
sed -i 's/ocnaddbackuprestore-serverprivatekey/ocnadd-
serverprivatekey/g' ocnadd_restore_occm.yaml
```

2. Run the following command to run the restore job:

```
kubectl create -f ocnadd_restore.yaml
```

Or, use the following command if OCCM is used:

```
kubectl create -f ocnadd_restore_occm.yaml
```

> **Note**
>
> Make sure to delete all the backup, restore, and verify jobs before creating the restore job. Related jobs are `ocnaddbackup`, `ocnaddrestore`, `ocnaddverify`, and `ocnaddmanualbackup`.

3. Wait for the restore job to be completed. It usually takes 10 to 15 minutes or more depending upon the size of the backup.
4. To restart the Redundancy Agent pods post OCNADD Restore, see Two-Site Redundancy Fault Recovery.
5. Perform the rollout restart for the deployments in management group and all the available worker groups (relay agent and mediation) in the provided order:
   a. Perform rollout restart for management group

   ```
   kubectl rollout restart deployment -n <mgmt-grp-namespace>
   ```

   b. Perform rollout restart for relay agent group

   ```
   kubectl rollout restart deployment -n <relay-agent-grp-namespace>
   ```

   c. Perform rollout restart for all mediation group(s) one after the other

   ```
   kubectl rollout restart deployment -n <mediation-grp-namespace>
   ```

> **Note**
>
> If the backup is not available for the mentioned date, the pod will be in an error state, notifying the backup is not available for the given date: $DATE. In such case, provide the correct backup dates and repeat the procedure.

8.8 Configuring Backup and Restore Parameters

To configure backup and restore parameters, configure the parameters listed in the following table:

Table 8-2 Backup and Restore Parameters

Parameter Name	Data Type	Range	Default Value	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
BACKUP_STORAGE	STRING	-	20Gi	M	Persistent Volume storage to keep the OCNADD backups
BACKUP_CRONEXPRESSION	STRING	-	0 8 * * *	M	Cron expression to schedule backup cronjob
BACKUP_ARG	STRING	-	ALL	M	KAFKA, DB, or ALL backup
BACKUP_DATABASES	STRING	-	ALL	M	Individual databases or all databases backup that need to be taken
PURGE_DAYS	INTEGER	-	7	M	The number of days after which the backup file will be purged

8.9 Two-Site Redundancy Fault Recovery

This section describes how to perform fault recovery of the OCNADD sites with Two-Site Redundancy enabled.

Scenario 1: When DB backup is available for both sites

1. Follow the generic recovery procedure based on the failure scenarios described in the section "Fault Recovery."
2. Use the respective site's backup during the restore procedure.
3. Once the recovery is completed, restart the Redundancy Agent pods of the Primary site and the Secondary site.

Scenario 2: When DB backup is not available on one of the mated sites

1. Access any one of the pods of the working site and run the below curl command to delete Redundancy Configuration:
   - `kubectl exec -it -n <namespace> <pod> -- bash`

For example:

```
kubectl exec -it -n ocnadd-deploy ocnaddmanagementgateway-xxxx -- bash
```

- `curl -k --cert-type P12 --cert /var/securityfiles/keystore/serverKeyStore.p12:$OCNADD_SERVER_KS_PASSWORD --location -X DELETE 'https://ocnaddconfiguration:12590/ocnadd-configuration/v1/tsr-configure/<workergroup name>?sync=false'`
  Where,

  `<workergroup-name>` is the logical name of the worker group. For example, wg1.

2. Follow the generic recovery procedure based on the failure scenarios described in the section Fault Recovery Scenarios.
3. Once the recovery is completed, restart the Redundancy Agent pods first on the Primary site, then on the Secondary site.
4. Re-create the Redundancy Configuration from the Primary UI.

> **Note**
>
> If the DB was lost on the Primary site and the user wants the Secondary site configuration to be restored on the Primary site, then set the **Way** to **Bidirectional** while creating the **Redundancy Configuration**.