Oracle® Communications

Cloud Native Core CNC Console Disaster Recovery Guide

Release 22.4.1
F74461-02
January 2023

Oracle Communications Cloud Native Core CNC Console Disaster Recovery Guide, Release 22.4.1

F74461-02

Contents

My Oracle Support

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

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

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

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

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

Acronyms

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

Table Acronyms

Acronyms	Definition
AD	Active Directory
ASM	Aspen Service Mesh
BSF	Binding Support Function
CNCC	Cloud Native Core Console
CNE	Cloud Native Environment
OCCNE	Oracle Communications Cloud Native Environment
CS	Common Service
CRUD Operations	CREATE, READ, UPDATE, DELETE
ECDSA	Elliptic Curve Digital Signature Algorithm
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity Access Management
KPI	Key Performance Indicator
M-CNCC	Manager CNC Console or M-CNCC (also known as mCncc) is a CNCC instance which manages local OCCNE common service(s) and remote Agent CNC Console (s) (A-CNCC). M-CNCC has two components: M-CNCC IAM and M-CNCC Core.
M-CNCC IAM	Manager CNC Console IAM or M-CNCC IAM (also known as mCncc Iam) is an IAM component of M-CNCC. M-CNCC IAM contains M-CNCC IAM Ingress Gateway and M-CNCC IAM back-end microservices.
M-CNCC Core	Manager CNC Console Core or M-CNCC Core (also known as mCncc Core) is a core component of M-CNCC that provides GUI and API access portal for accessing NF and OCCNE common services. M-CNCC Core contains M-CNCC Core Ingress Gateway and M-CNCC Core back-end microservices.
A-CNCC	Agent CNC Console is a CNCC Core instance which manages local NF(s) and local OCCNE common services(s). A-CNCC is managed by M-CNCC. A-CNCC contains A-CNCC Core Ingress Gateway and A-CNCC Core back-end microservices. A-CNCC has no IAM component. A-CNCC is also known as A-CNCC Core or aCncc Core.
A-CNCC site	Site hosting A-CNCC
M-CNCC site	Site hosting M-CNCC
mTLS	Mutual Transport Layer Security
Instance	OCNF or OCCNE common service managed by either M-CNCC Core or A-CNCC Core.
Site	Kubernetes Cluster

Table (Cont.) Acronyms

Acronyms	Definition
CS	OCCNE Common Services like Grafana, Kibana, Jaeger, Prometheus, Alertmanager and so on.
MC	Multi Cluster. In multi cluster, a single CNCC can manage NF instances that accessess different Kubernetes clusters.
MO	Mananged Objects
MOS	My Oracle Support
LDAP	Lightweight Directory Access Protocol
LDAPS	Lightweight Directory Access Protocol (Over SSL)
NRF	Network Repository Function
OCNF	Oracle Communications Network Function
OSDC	Oracle Software Delivery Cloud
OSO	Operations Services Overlay
REST API	Representational State Transfer Application Programming Interface
SCP	Service Communication Proxy
SAML	Security Assertion Markup Language
SEPP	Security Edge Protection Proxy
TLS	Transport Layer Security
UDR	Unified Data Repository
UE	User Equipment
URI	Subscriber Location Function

What's New in This Guide

This section introduces the documentation updates for Release 22.4.x in Oracle Communications Cloud Native Core Console Disaster Recovery Guide.

Release 22.4.1- F74461-02, January 2023

No updates are made in this document for Release 22.4.1.

Release 22.4.0- F74461-01, November 2022

No updates are made in this document for Release 22.4.0.

1 Introduction

This document describes procedures to perform disaster recovery for Oracle Communications Cloud Native Core Console (CNC Console) deployment. The operators can take database backup and restore them either on the same or a different cluster.

Prerequisites

Before you run any disaster recovery procedure, ensure that the following prerequisites are met:

- DBTier should be in healthy state and available on multiple sites along with CNC Console.
- On demand backup should be enabled on DBTier. Scheduled regular backups help to:
  - Restore stable version of the CNC Console configuration database.
  - Minimize significant loss of data due to upgrades or roll back failures.
  - Minimize loss of data due to system failure.
  - Minimize loss of data due to data corruption or deletion due to external input.
  - Migrate Console database information from one site to another.
- Custom values file used at the time of Console deployment is retained. If the `custom_values.yaml` file is not retained, then regenerate it manually. This task increases the overall disaster recovery time.
- Docker images used during the last installation or upgrade must be retained in the external data source.
  For the CNC Console database backup and restore prerequisites, see CNC Console Database Backup and Restore section.

Audience

The CNC Console Disaster Recovery Guide provides guidance to network administrators and professionals responsible for deployment and maintenance of CNC Console.

References

While performing disaster recovery procedures, you may refer to the procedures defined in the following documents:

- *Oracle Communications CNC Console Installation and Upgrade Guide*
- *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide*
- *Oracle Communications Cloud Native Environment (OCCNE) Installation Guide*

2
Disaster Recovery Impacted Area

The following table describes scenarios about the impacted areas during CNC Console disaster recovery:

Table 2-1 Disaster Recovery Scenarios Impact Information

Scenario	Requires Disaster Recovery or re-install of CNE?	Requires Disaster Recovery or re-install of cnDBTier?	Requires Disaster Recovery or re-install of CNC Console?	Comments
Scenario 1: Complete Site Failure	Yes	Yes	Yes	
Scenario 2: cnDBTier Corruption	No	Yes	No	DBTier must be restored from backup. Re-install of CnDBTier can be considered, if restoring from backup is not possible. Use helm upgrade if DB configuration is changed
Scenario 3: Console Configuration Database Corruption	No	No	No	Backup and restore of configuration database is required on the impacted site. This needs periodic backup. (Applicable for M-CNCC IAM)
Scenario 4: Deployment Failure	No	No	Yes	CNC Console DB is not restored. Only helm uninstall/install is done. (Applicable for M-CNCC IAM, M-CNCC Core and A-CNCC Core)

Table 2-1 (Cont.) Disaster Recovery Scenarios Impact Information

Scenario	Requires Disaster Recovery or re-install of CNE?	Requires Disaster Recovery or re-install of cnDBTier?	Requires Disaster Recovery or re-install of CNC Console?	Comments
Scenario 5: NF Instance Failure	No	No	Yes	The NF endpoints must be updated in the custom values.yaml file and helm upgrade must be performed to incorporate the latest endpoints (Applicable at M-CNCC Core and A-CNCC Core).

3
Disaster Recovery Scenarios

This chapter describes disaster recovery scenarios of CNC Console and how to recover from those scenarios.

Scenario 1: Complete Site Failure

This section describes how to perform disaster recovery when either one, many, or all of the sites have software failure.

The following are site failure scenarios:

- Single or Multiple Site Failure
- All Sites Failure

Scenario 1A: Single or Multiple Site Failure

This scenario applies when one or more sites, and not all sites, have failed and there is a requirement to perform disaster recovery. It is assumed that the user has DBTier and Oracle Communications CNC Console installed on multiple sites with automatic data replication and backup enabled.

To recover the failed sites:

1. Run the Cloud Native Environment (CNE) installation procedure to install a new cluster. For more information, see *Oracle Communications Cloud Native Environment (OCCNE) Installation Guide.*
2. For DBTier disaster recovery:
    a. Take on-demand backup from the mate site that has health replication with the failed site or sites. For more information about on-demand backup, see the Create On-demand Database Backup chapter in the *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide.*
    b. Use the backup data from the mate site to restore the database. For more information about database restore, see the Restore Georeplication Failure chapter in the *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide.*
3. Install CNC Console helm chart. For more information about installing CNC Console, see the Installing CNC Console chapter in the *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide.*

Scenario 1B: All Sites Failure

This scenario applies when all sites have failed and there is a requirement to perform disaster recovery. It is assumed that the user has DBTier and Oracle Communications CNC Console installed on multiple sites with automatic data replication and backup enabled.

To recover all the failed sites:

1. Run the Cloud Native Environment (CNE) installation procedure to install a new cluster. For more information, see *Oracle Communications Cloud Native Environment (OCCNE) Installation Guide.*
2. Use an on-demand backup file to restore the database from previous data backup. For more information about database restore, see the Restore Georeplication Failure chapter in the *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide.*

> **Note:**
>
> - The auto-data backup file is one that is built from scheduled automatic backup.
> - The Restore Georeplication Failure chapter contains a procedure for two sites where one of the clusters has a fatal error. You can perform the same procedure for all the sites in a multiple site setup.

3. Install CNC Console helm chart. For more information about installing CNC Console, see the Installing CNC Console chapter in the *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide.*

Scenario 2: cnDBTier Corruption

This section describes how to recover a database when the data replication has failed due to database corruption and cnDBTier has failed in single, multiple sites, or all sites.

When the database gets corrupted, the database on all the other sites can also get corrupted due to data replication. It depends on the replication status after the corruption has occurred. If the data replication fails due to database corruption, then DBTier fails in either single or multiple sites (not all sites). If the data replication is successful, then database corruption replicates to all the cnDBTier sites and cnDBTier fails in all sites.

The following are cnDBTier failure scenarios:

If corrupted database is replicated to mated sites, follow:

- When cnDBTier fails in Single or Multiple (but not all) Sites

If corrupted database causes replication failure and hence local to a site, follow:

- When cnDBTier fails in all Sites

> **Note:**
>
> This impacts all the NFs using the corrupted cnDBTier. All the NFs sharing cnDBTier needs to do a disaster recovery as cnDBTier is corrupted.

Scenario 2A: When DBTier fails in Single or Multiple (but not all) Sites

This section describes how to recover database when the data replication has failed due to database corruption and DBTier has failed in either single or multiple sites (not all sites).

To recover database:

1. Uninstall CNC Console helm chart. For information about uninstalling CNC Console, see the Uninstalling CNC Console chapter in the *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide.*
2. For DBTier disaster recovery:
   a. Create on-demand backup from mated site that has health replication with failed site. For more information about DBTier backup, see the Create On-demand Database Backup chapter in the *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide*.
   b. Use the backup data from mate site for restoration. For more information about DBTier restore, see the Restore Georeplication Failure chapter in *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide.*

   > **Note:**
   >
   > The Restore Georeplication Failure chapter in *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide*. has a procedure for two sites where one of the cluster has fatal error. You can perform that procedure for all the sites in a multiple site setup.

3. Install CNC Console helm chart. For more information about installing CNC Console, see the Installing CNC Console chapter in the Oracle Communication Cloud Native Core Console Installation and Upgrade Guide.

Scenario 2B: When DBTier failed in all Sites

This section describes how to recover database when successful data replication corrupts all the DBTier sites.

To recover database:

1. Uninstall CNC Console helm charts. For more information about uninstalling CNC Console, see the Uninstalling CNC Console chapter in the *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide.*
2. For DBTier disaster recovery:
   a. Use an on-demand backup file to restore the database from the previous data backup. For more information about DBTier restore, see the Restore Georeplication Failure chapter in the *Oracle Communications Cloud Native Core DBTier Disaster Recovery Guide.*

> **Note:**
>
> The Restore Georeplication Failure chapter has a procedure for two sites where one of the cluster has fatal error. You can perform that procedure for all the sites in a multiple site setup.

3. Install CNC Console helm charts. For more information about installing CNC Console, see the Installing CNC Console chapter in the *Oracle Communication Cloud Native Core Console Installation and Upgrade Guide.*

Scenario 3: Console Configuration Database Corruption

This section describes how to recover when the CNC Console configuration database is corrupted.

For recovery and restore procedure, see Disaster Recovery Procedures - DB Backup and Restore section.

Scenario 4: Deployment Failure

This section describes how to recover CNC Console when its deployment fails.

For recovery and restore procedure, see Restoring CNC Console.

Scenario 5: NF Instance Failure

Perform this procedure to recover from NF instance failure.

1. Refer to the procedures in the specific NF Disaster Recovery Guide to find the necessary action required to be taken.
2. Check whether the NF endpoints are the same.
   a. If the NF endpoints are the same, no change is required in CNC Console side.
   b. If the NF endpoints are different, update the NF IP and Port in the *occncc_custom_values_<version>.yaml* file and perform a helm upgrade operation to incorporate the new NF URL.

A

Disaster Recovery Procedures - DB Backup and Restore

Introduction

Perform this procedure to take a backup of the Console database (DB) and restore the database on a different cluster. This procedure is for on-demand backup and restore of Console DB. The commands used for these procedures are provided by the MYSQL Network Database (NDB) cluster.

Prerequisite

Ensure that the MYSQL NDB cluster is in a healthy state, and each database node of it should be in the running state. Run the following command to check the status of cnDBTier service:

```
kubectl -n <namespace> exec <management node pod> -- ndb_mgm -e show
```

Where,

- `<namespace>` is the namespace where cnDBTier is deployed
- `<management node pod>` is the management node pod of cnDBTier
- In case of cnDBTier to verify the prerequisites, check whether the mysql pod is up and running.

Console DB Backup

If the Console database backup is required, do the following:

1. Log in to any of the SQL node or API node, and then run the following command to take dump of the database:

   ```
   kubectl exec -it <sql node> -n <namespace> bash
   mysqldump --quick -h127.0.0.1 -u <username> -p  <databasename>| gzip >
   <backup_filename>.sql.gz
   ```

   Where,

   - `<sql node>` is the SQL node of cnDBTier
   - `<namespace>` is the namespace where cnDBTier is deployed
   - `<username>` is the database username.
   - `<databasename>` is the name of the database that has to be backed up
   - `<backup_filename>` is the name of the backup dump file
2. Enter the Console database name and password in the command when prompted.

Example:

```
kubectl exec -i -n occne-ndb ndbmysqld-0 -- mysqldump --single-
transaction --no-tablespaces --no-create-info
      -h 127.0.0.1 -u cnccuser -p cnccdb | gzip >
cnccdbBackup.sql.gz
```

> **Note:**
>
> Ensure that there is enough space on the directory to save the backup file.

Console Restore Procedure

If the Console database restore is required, do the following:

1. Drop existing database and recreate database.
2. Restore this new database with the DB Schema file provided as part of package.
3. Unzip the cnccdbBackup.sql.gz file.
4. Rearrange the back up sql file in correct order using given procedure
5. Populate the DB with rearranged SQL file.
6. Log in to the deployment cluster, drop the existing database and create a new database. Now restore this new database with the DB Schema file provided as part of package.(`occncc_rollback_iam_schema_<version>.sql`).
7. Create database, database user, and grant permissions as described in *Oracle Communication Cloud Native Core Console Installation and Upgrade Guide* . Command:

```
DROP DATABASE <CNCC Database>
CREATE DATABASE IF NOT EXISTS <CNCC Database>;
GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, REFERENCES,
INDEX, CREATE TEMPORARY TABLES, DELETE, UPDATE, EXECUTE ON <M-CNCC
IAM Database>.* TO'<CNCC IAM DB User Name>'@'%';
```

Example:

```
To be executed in the mysql pod:
DROP DATABASE cnccdb;
CREATE DATABASE IF NOT EXISTS cnccdb;
GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES, REFERENCES,
INDEX, CREATE TEMPORARY TABLES, DELETE, UPDATE, EXECUTE ON
cnccdb .* TO'cnccusr'@'%';
```

> **Note:**
>
> The database name created in this step should be the same as the database name created in the next sub step. Also, the Kubernetes secret should be the same as in the `values.yaml` file used for installing Console.

8. To restore the database to the new database created, run the following command:

```
kubectl exec -i -n <namespace> <podname> -- mysql  -h 127.0.0.1 -u
<username> -p<password> <CNCC Database name> < <backup_filename>
```

Example:

```
kubectl exec -i -n cndbtier1 ndbmysqld-0 -- mysql  -h 127.0.0.1 -u
cnccusr -pcnccpasswd cnccdb_site1_cluster1 < create-schema-NDB.sql
```

Procedure to Re-arrange the DB Dump file

The following procedure is to re-arrange the DB Dump file sequentially so that the user wont get any foreign key constraints issue. To do that ENV variables must be created and run it through a for loop.

1. Run the following command to convert the mysqldump file which was taken as a backup (sql.gz file) to a sql file:

```
gunzip -d <filename.sql.gz>
```

Example:

```
gunzip -d cnccdbBackup.sql.gz
```

2. Run the following command to create an ENV which has the sequential table order required:

```
export KC_TABLES="ADMIN_EVENT_ENTITY RESOURCE_SERVER
RESOURCE_SERVER_POLICY
                                    ASSOCIATED_POLICY REALM CLIENT
AUTHENTICATION_FLOW
                                    AUTHENTICATION_EXECUTION
AUTHENTICATOR_CONFIG
                                    AUTHENTICATOR_CONFIG_ENTRY
BROKER_LINK CLIENT_ATTRIBUTES
                                    CLIENT_AUTH_FLOW_BINDINGS
KEYCLOAK_ROLE CLIENT_INITIAL_ACCESS
                                    CLIENT_NODE_REGISTRATIONS
CLIENT_SCOPE CLIENT_SCOPE_ATTRIBUTES
                                    CLIENT_SCOPE_CLIENT
CLIENT_SCOPE_ROLE_MAPPING USER_SESSION
                                    CLIENT_SESSION
CLIENT_SESSION_AUTH_STATUS CLIENT_SESSION_NOTE
                                    CLIENT_SESSION_PROT_MAPPER
CLIENT_SESSION_ROLE
```

```
                                        CLIENT_USER_SESSION_NOTE
COMPONENT COMPONENT_CONFIG
                                        COMPOSITE_ROLE
DATABASECHANGELOG USER_ENTITY CREDENTIAL
                                        DATABASECHANGELOGLOCK
DEFAULT_CLIENT_SCOPE EVENT_ENTITY
                                        FEDERATED_IDENTITY
FEDERATED_USER FED_USER_ATTRIBUTE
                                        FED_USER_CONSENT
FED_USER_CONSENT_CL_SCOPE FED_USER_CREDENTIAL
                                        FED_USER_GROUP_MEMBERSHIP
FED_USER_REQUIRED_ACTION
                                        FED_USER_ROLE_MAPPING
KEYCLOAK_GROUP GROUP_ATTRIBUTE
                                        GROUP_ROLE_MAPPING
IDENTITY_PROVIDER IDENTITY_PROVIDER_CONFIG
                                        IDENTITY_PROVIDER_MAPPER
IDP_MAPPER_CONFIG MIGRATION_MODEL
                                        OFFLINE_CLIENT_SESSION
OFFLINE_USER_SESSION POLICY_CONFIG
                                        PROTOCOL_MAPPER
PROTOCOL_MAPPER_CONFIG REALM_ATTRIBUTE
                                        REALM_DEFAULT_GROUPS
REALM_LOCALIZATIONS
                                        REALM_ENABLED_EVENT_TYPES
REALM_EVENTS_LISTENERS
                                        REALM_REQUIRED_CREDENTIAL
REALM_SMTP_CONFIG
                                        REALM_SUPPORTED_LOCALES
REDIRECT_URIS REQUIRED_ACTION_CONFIG
                                        REQUIRED_ACTION_PROVIDER
RESOURCE_SERVER_RESOURCE
                                        RESOURCE_ATTRIBUTE
RESOURCE_POLICY RESOURCE_SERVER_SCOPE
                                        RESOURCE_SCOPE
RESOURCE_SERVER_PERM_TICKET RESOURCE_URIS
                                        ROLE_ATTRIBUTE SCOPE_MAPPING
SCOPE_POLICY USERNAME_LOGIN_FAILURE
                                        USER_ATTRIBUTE USER_CONSENT
USER_CONSENT_CLIENT_SCOPE
                                        USER_FEDERATION_PROVIDER
USER_FEDERATION_CONFIG
                                        USER_FEDERATION_MAPPER
USER_FEDERATION_MAPPER_CONFIG
                                        USER_GROUP_MEMBERSHIP
USER_REQUIRED_ACTION USER_ROLE_MAPPING
                                        USER_SESSION_NOTE WEB_ORIGINS";
```

3. Create an ENV pointing to the sql file to be filtered:

```
export KC_BACKUP="./<Backup SQL Dump File>";
```

4. Example:

```
export KC_BACKUP="./cnccdbBackup.sql";
```

5. Run the following command to re-arrange the dump file which you have taken earlier to ensure hat it is in sequential insertion order by running the following for loop command:

```
for i in $KC_TABLES; do grep "INSERT INTO \`$i\`" $KC_BACKUP; done >
<file name along with its location>
```

   Example:

```
for i in $KC_TABLES; do grep "INSERT INTO \`$i\`" $KC_BACKUP; done > /tmp/
restore.sql
```

6. Run the following command to populate the DB with data using the file which you have right now after filtering the sqldump file:

```
kubectl exec -i -n <namespace> <podname> -- mysql -h 127.0.0.1 -u
<username> -p<password><CNCC Database name> < <backup_filename>
```

7. Example:

```
kubectl exec -i -n cndbtier1 ndbmysqld-0 -- mysql -h 127.0.0.1 -u cnccusr
-pcnccpasswd cnccdb <restore.sql
```

> **Note:**
>
> If multiple sites are corrupted, perform the restore procedure on all sites.

B
Restoring CNC Console

Perform this procedure to restore CNC Console.

Prerequisites:

- Take a backup of the `occncc_custom_values_<version>.yaml` file that was used for installing CNC Console.
- Take a backup of the CNC Console database and restore the database as described in Disaster Recovery Procedures - DB Backup and Restore section.

Corruption in CNC Console Deployment

This section describes how to recover the CNC Console when the deployment is corrupted.

Console Deployment Corruption

If Console deployment is corrupted, do the following:

1. Run the following command to uninstall the corrupted CNCC deployment deployment:

   ```
   helm uninstall <deployment name> --namespace <namespace>
   ```

   Where,

   - `<deployment name>` is a name used to track this installation instance.

   Example:

   ```
   helm uninstall cncc --namespace cncc
   ```

2. Install Console using the backed up copy of the `occncc_custom_values_<version>.yaml` file.
   For information about installing CNC Console using Helm, see *Oracle Communication Cloud Native Core Console Installation and Upgrade Guide*.