Oracle® Communications

Cloud Native Configuration Console Troubleshooting Guide

Release 23.4.4

F87108-05

December 2024

Oracle Communications Cloud Native Configuration Console Troubleshooting Guide, Release 23.4.4

F87108-05

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 terminologies used in the document:

Table Acronyms

Acronym	Definition
AD	Active Directory
ASM	Aspen Service Mesh
BSF	Oracle Communications Cloud Native Core, Binding Support Function
cnDBTier	Oracle Communications Cloud Native Core, cnDBTier
CNC Console	Oracle Communications Cloud Native Configuration Console
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
CS	Common Service
CRUD Operations	CREATE, READ, UPDATE, DELETE
OCNADD	Oracle Communications Network Analytics Data Director
ECDSA	Elliptic Curve Digital Signature Algorithm
EIR	Equipment Identity Register
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 CNE 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 Core	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. A-CNCC has no IAM component. A-CNCC is also known as A-CNCC Core or aCncc Core.
M-CNCC Kubernetes cluster	Kubernetes cluster hosting M-CNCC
mTLS	Mutual Transport Layer Security
OCNWDAF	Oracle Communications Networks Data Analytics Function

Table (Cont.) Acronyms

Acronym	Definition
Instance	NF or CNE common service managed by either M-CNCC Core or A-CNCC Core.
Site	Kubernetes Cluster
CS	CNE 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	Oracle Communications Cloud Native Core, Network Repository Function
OCNF	Oracle Communications Network Function
OSDC	Oracle Software Delivery Cloud
OSO	Oracle Communications Operations Services Overlay
PROVGW	Provisioning Gateway
REST API	Representational State Transfer Application Programming Interface
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy
SAML	Security Assertion Markup Language
SBA	Service Based Architecture
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy
TLS	Transport Layer Security
UDR	Oracle Communications Cloud Native Core, Unified Data Repository
UE	User Equipment
URI	Subscriber Location Function
SSO	Single Sign On

What's New in This Guide

This section lists the documentation updates for Release 23.4.x in Oracle Communications Cloud Native Configuration Console Troubleshooting Guide.

Release 23.4.4 - F87108-05, December 2024

There are no updates in this release.

Release 23.4.2 - F87108-04, October 2024

Added the Updating M-CNCC IAM KC log level to the Types of Logs section.

Release 23.4.2 - F87108-03, July 2024

There are no updates in this release.

Release 23.4.1 - F87108-02, April 2024

There are no updates in this release.

Release 23.4.0 - F87108-01, December 2023

- Added details on troubleshooting the following scenario:
  - How to Reset CNC Console IAM Admin Password

1 Introduction

This document provides information about troubleshooting Oracle Communications Cloud Native Configuration Console (CNC Console).

1.1 Overview

CNC Console is a single screen solution to configure and manage Network Functions (NFs). The CNC Console has the following two modules:

- CNC Console Core: CNC Console Core acts as GUI or API portal for NFs and CNE common services. CNC Console Core module includes CNC Console and its integration with other Cloud Native Core network functions. The CNC Console provides user interface that can be used to configure parameters for the following CNC network functions:
  - Oracle Communications Cloud Native Core, Binding Support Function (BSF)
  - Oracle Communications Cloud Native Core, Service Communication Proxy (SCP)
  - Oracle Communications Cloud Native Core, Network Repository Function (NRF)
  - Oracle Communications Cloud Native Core, Converged Policy (Policy)
  - Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP)
  - Oracle Communications Cloud Native Core, Unified Data Repository (UDR)
  - Oracle Communications Cloud Native Core Network Slice Selection Function (NSSF)
  - CNE Common Services
  - Data Director (DD)
  - Oracle Communications Network Data Analytics Function (NWDAF)
  - Provisioning Gateway (PROVGW)
  - Oracle Communications Cloud Native Core, Certificate Management (OCCM)
- CNC Console Identity Access Management (CNCC IAM): CNCC IAM acts as local identity provider and broker for external identity provider. CNCC IAM module includes the required authentication and authorization procedures such as creating and assigning roles to users.

1.2 Reference

Following are the reference documents:

- *Oracle Communications Cloud Native Configuration Console Installation and Upgrade Guide*
- *Oracle Communications Cloud Native Core DBTier Installation ans Upgrade Guide*
- *Oracle Communications Cloud Native Configuration Console User Guide*
- *Oracle Communication Cloud Native Core Data Collector Guide*

2
Logs

This chapter explains the process to retrieve the logs and status that can be used for effective troubleshooting.

2.1 Collecting Logs

This section describes the steps to collect logs from PODs and containers. Perform the following steps:

1. Run the following command to get the PODs details:

   ```
   $ kubectl -n <namespace_name> get pods
   ```

2. Collect the logs from the specific pods or containers:

   ```
   $ kubectl logs <podname> -n <namespace> -c <containername>
   ```

3. Store the log in a file using the following command:

   ```
   $ kubectl logs <podname> -n <namespace>  > <filename>
   ```

4. (Optional) You can also use the following commands for the log stream with file redirection starting with last 100 lines of log:

   ```
   $ kubectl logs <podname> -n <namespace> -f --tail <number of lines> >
   <filename>
   ```

For more information on kubectl commands, see Kubernetes website.

2.2 Log Formats

This section provides information about the log formats.

Log4j JSON Format

CNCC Message Format

Log4j JSON Format

Following are the log format and fields. All logs are represented in JSON format.

```
{
     "thread": <threadName>,
     "level": <log_level>,
     "loggerName": <name_of_the_logging_class>,
     "message": <message>,
     "instant": <timestamp_in_miliseconds>,
     "logTimestamp": <timestamp_in_readable_format>,
     "threadId": <threadId>,
```

```
    "threadPriority": <threadPriority>,
    "pod": <name_of_the_pod>,
    "processId": <processId>,
    "contextMap": <context_map>,
    "ocLogId": <unique_trace_id_for_every_request>,
    "instanceType": <instanceType>,
    "ingressTxId": <IngressTransactionId>
}
```

Table 2-1 Log Details

Name	Description	Example
thread	Name of the thread	"thread": "reactor-http-epoll-1"
level	Level of the log. It can be: Log level (INFO, WARN, DEBUG, TRACE)	"level": "INFO"
loggerName	Name of the class that generated the log	"loggerName": "ocpm.cne.gateway.cncc.GatewayApplication"
logTimestamp	Time represented in human readable format and in UTC. Format is *date:yyyy-MM-dd'T'HH:mm:ss.SSSZ* EFK friendly and also follows Oracle Standards	"logTimestamp": 2020-07-04'T'12:00:40.702Z
message	Information about the event	"message": "Started Application....." By default, all messages are in simple string except ***Audit Log***, ***Security Log*** which are represented in CNCC Message Format.
instant	The Date and Time the event occurred in epoch second and nano seconds	"instant": { "epochSecond": 1590045388, "nanoOfSecond": 339789000}
processId	Linux process Identifier (for a multi-process host)	Linux process Identifier (for a multi-process host).
threadId	Id of the thread	"threadId":"43"
threadPriority	Priority assigned to the thread	"threadPriority": 5
pod	Name of the pods where the log is generated	"cncc-mcore-ingress-gateway-77df795fb5-wv2sb"
contextMap	It holds information added to threadContext.	"contextMap": { "hostname": "cncc-mcore-ingress-gateway-77df795fb5-wv2sb", "ingressTxId": "ingress-tx-1460885598"}
ocLogId	It contains the trace id that is uniquely generated for every request of the format "<timestamp(in milliseconds)>_<thread Id>_<POD name>"	It contains the trace id that is uniquely generated for every request of the format "<timestamp (in milliseconds)>_<thread Id>_<POD name>"
instanceType	Static tag which implies that instance type is production	"instanceType": "prod"
ingressTxId	It contains id of the format "ingress-tx-<Random no>" to track every transaction	"ingressTxId": ingress-tx-1904660570

CNCC Message Format

Table 2-2 CNCC Message Format

Name	Description	Example	Possible Values
logType	Indicates whether it is ***Security Log*** or ***Audit Log***	logType=AUDIT	AUDIT SECURITY
type	Indicates nature or action of the log	type=REQUEST	For Security Log: REQUEST, RESPONSE For Audit Log: LOGIN, ACCESS_RESOURCE, ACCESS_RESOURCE_ERROR, LOGOUT
resourceType	Indicates what is the resource being requested for	resourceType=SCP	CM_SERVICE (For default route) CNCC (For User Login Activity) SCP UDR NRF PCF (all CNCC supported NFs)
userId	Id of the user who triggered request or action	userId=3314f54f-08bf-489d-b395-27bf56da1262	NA
username	Name of the user	username= "user1"	NA
status	HTTP status of the response.	status=200 OK	NA
operationType	HTTP method of the request	operationType=GET	NA
scheme	Indicates the scheme of the request	scheme=http	NA
remoteAddress	The remote address that is associated with the request. It also means the remote address to where this request is connected when available.	remoteAddress=/192.168.219.64:53587	NA
localAddress	The local address that is associated with the request. It also means the local address to where this request is connected when available.	localAddress=cncc-mcore-ingress-gateway.cncc.svc.cluster.local/<unresolved>:30075	NA
resourcePath	Request uri	resourcePath=/soothsayer/v1/canaryrelease/	NA
queryParams	Query parameters associated with request	queryParams={form_id=9, page=1, view_id=78}	NA

Table 2-2 (Cont.) CNCC Message Format

Name	Description	Example	Possible Values
headers	Headers associated with request or response	headers={Accept=*/*, X-Requested-With=XMLHttpRequest, User-Agent=Mozilla/5.0 (Windows NT 10.0; WOW64; rv:68.0) Gecko/20100101 Firefox/68.0, Connection=keep-alive, Host=cncc-core-ingress-gateway.cncc.svc.cluster.local:30075, Accept-Language=en-US,en;q=0.5, Accept-Encoding=gzip, deflate, DNT=1, Content-Type=application/json; charset=utf-8}	NA
payload	Payload or Data associated with request or response	payload=[{"serviceName":"n5g-eir-eic","canaryReleaseFlag":true,"apiFullVersion":"2.0.0","canaryTraffic":5}	NA
authenticationType	This indicates whether user is requesting resource logged in using CNCC or directly accessing through postman or curl.	authenticationType=OAUTH	OAUTH -> User is logged in through CNC Console application and accessing resource JWT -> User is accessing resource directly through postman or curl

2.3 Log Levels

Logs register system events along with their date and time of occurrence. They also provide important details about a chain of events that could have led to an error or problem.

A log level helps in defining the severity level of a log message. For CNC Console, the log level of a microservice can be set to any one of the following valid values:

- **TRACE:** A log level that describes events, as a step by step execution of code. This can be ignored during the standard operation, but may be useful during extended debugging sessions.
- **DEBUG:** A log level used for events during software debugging when more granular information is needed.
- **INFO:** A standard log level indicating that something has happened, an application has entered a certain state, etc.
- **WARN:** A log level indicates that something unexpected has happened in the application, a problem, or a situation that might disturb one of the processes. But this does not mean that the application has failed. The WARN level should be used in situations that are unexpected, but the code can continue to work.
- **ERROR:** A log level that should be used when an application hits an issue preventing one or more functionalities from functioning.

Using this information, the logs can be filtered based on the system requirements. For instance, if you want to filter the critical information about your system from the informational log messages, set a filter to view messages with only WARN log level in Kibana.

2.4 Types of Logs

The CNC Console logs can be categorized into following types:

- Regular logs
- Audit logs
- Security logs

Regular logs

These logs contain error messages, warnings, or other events written within the application that provide logical, high level information about the application and ongoing events.

Example:

```
{"level": "INFO","message": "Started GatewayApplication in 10.748 seconds
(JVM running for 12.825)"}
{"level": "INFO","message": "Creating plain httpClient"}
{"level": "INFO","message": "Creating plain restTemplate"}
{"level": "ERROR","message": "Can't get cfgs of topic
public.dynamic.datamodel,  exception is:\n
javax.ws.rs.ProcessingException: java.net.ConnectException: Connection
refused (Connection
        refused)"}
```

Audit Logs

These logs contain user related information and the activity within the system.

The following events are logged in CNC Console Core:

- **Log in**: A user has logged in.
- **Access Resource**: A user is accessing a particular NF resource.
- **Access Resource Error**: A user is denied from accessing a particular NF resource.
- **Logout**: A user has logged out.

> **Note**
>
> The user can find the CNCC Core User Activity logs as part of *cncc-core-ingress-gateway* and are represented in CNCC message format.

The following events are logged in CNCC IAM:

Login events

- **Log in:** An admin user has logged in.
- **Register:** An admin user has registered.

- **Logout:** An admin user has logged out.
- **Code to Token:** An application or a client has exchanged a code for a token.
- **Refresh Token:** An application or a client has refreshed a token.

Account events

- Update Email: The email address for an account has changed.
- Update Profile: The profile for an account has changed.
- Send Password Reset: A password reset email has been sent.
- Update Password: The password for an account has changed.

> **Note**
>
> The user can find the CNCC IAM User Activity logs as part of *cncc-iam-0*, represented in Keycloak format. These events are provided by keycloak and documented under Keycloak Auditing End Events.
> Logging Error Logs are recorded by keycloak container as :

```
^[[0m^[[33m10:12:57,388 WARN  [org.keycloak.events] (default
task-3) type=LOGIN_ERROR, realmId=master, clientId=security-admin-
console,
      userId=ef58d62e-a0a8-4f4e-bcc6-abccf917641c,
ipAddress=192.168.203.108, error=invalid_user_credentials,
auth_method=openid-connect, auth_type=code,
      redirect_uri=http://10.75.240.33:30085/cncc/auth/admin/master/
console/,
      code_id=3e6d822a-9e82-4660-bb01-a814f7ae8f97, username=admin,
      authSessionParentId=3e6d822a-9e82-4660-bb01-a814f7ae8f97,
    authSessionTabId=2ak6Xwal-28
```

Security Logs

The security logs contain the header, payload, method, scheme, URI information for all the requests and corresponding responses.

Disabling Security Logs

By default **Security Log** will be enabled for *M-CCNC IAM*, *M-CNCC Core* and *A-CNCC Core*. You can disable this by setting *securityLogEnabled* flag to **false** in *custom values.yaml* file.

```
# CNCC configuration
cncc:
  # Enable security logs for CNCC
  securityLogEnabled: false
```

Header Information

At all the log levels, sensitive information like **Cookies** are masked.

Note

The user can find the Security logs :

- For M-CNCC Core and A-CNCC Core, these are logged as part of *cncc-mcore-ingress-gateway or cncc-acore-ingress-gateway* and are represented in CNCC message format.
- For M-CNCC IAM, these are logged as part of *cncc-iam-ingress-gateway* and are represented in CNCC message format.

Log Levels

Default log levels set for M-CNCC Core and A-CNCC Core:

```
ingress-gateway:
  log:
    level:
      cncc:
        root: WARN
        audit: INFO
        security: INFO
```

Default log levels set for M-CNCC IAM:

```
ingress-gateway:
  log:
    level:
      cncc:
        root: WARN
        security: INFO
```

Updating M-CNCC IAM KC log level

- By default the log level of M-CNCC IAM KC is set to **WARN,org.keycloak.events:DEBUG**
  - This means the *root log-level* for is set to **WARN** and the *org.kecyalok.events* package is set to **DEBUG**
- In the **level** label set the log level, follow the below instruction about the options available to set log level
  - **Log level**
    - TRACE
    - DEBUG
    - INFO
    - WARN
    - ERROR
    - FATAL
  - **level: INFO**
    INFO, WARN, ERROR, FATAL will be covered for logs of all packages in KC

  - **level: INFO, DEBUG**
    In this case it will take the last one as log level, here DEBUG
  - **level: INFO, package-group: WARN**
    What does this mean ?

    INFO level logs for all, but only for that particular package log level will be warn, for example:

    ```
    level: INFO,org.keycloak:WARN,org.infinispan:WARN,org.jgroups:TRACE
    ```

  - **level: OFF**
    No logs will appear
  - **level: ALL**
    logs of all the level will appear in the logs
- Sample M-CNCC IAM KC log configuration:

  ```
  kc:
    log:
      level: WARN,org.keycloak.events:DEBUG
  ```

Supported Headers for Logging

Header	Header values (regex)
Content-Type	^application/x-www-form-urlencoded.*
	^application/json.*
	^application/problem+json.*
Accept	^application/json.*
	^application/ld+json.*
	^application/xml.*
	^multipart/form-data.*

Role of supporting headers in CNCC Audit and Security logs

- At INFO level, only those request and response that match the supporting headers and values are logged.
- At DEBUG level, no supporting headers used and all request and response are logged.
- At ERROR / WARN, no supporting headers used and only error or warnings are logged.

> ⓘ **Note**
>
> Any failure in authorizing a request will always be logged irrespective of the supported header configuration.

2.5 Configuring Security Logs

This section provides the details about configuring security logs.

Setting at Log Level

By default, Security Log is set to the "INFO" level for both CNC Console Core and CNC Console IAM. You can change the log level by setting log.level.cncc.security to the required level in core and iam values.yaml file.

values.yaml

```
#Set the root log level
log:
  level:
    root: WARN
    ingress: INFO
    oauth: INFO
    cncc:
      security: INFO
```

Disabling Security Log
By default, the Security Log is enabled for both CNCC Core and CNCC IAM. You can disable this by setting `securityLogEnabled` flag to false in core and iam values.yaml file.

values.yaml

```
# CNCC configuration
cncc:
  enabled: false
  enablehttp1: false
  securityLogEnabled: false
```

2.5.1 Accessing logs

This section gives information about how to access the logs.

The CNCC application logs can be accessed in following ways:

1. Run the following command to view logs of a cncc application pod:

   ```
   kubectl logs -f -n <cncc_namespace> <pod_name> -c <container_name>
   ```

   Example:

   CNCC Core:

   ```
   $ kubectl logs -f -n cncc cncc-mcore-ingress-gateway-77df795fb5-wv2sb -c
   ingress-gateway (Security & Audit Log)
   ```

   CNCC IAM:

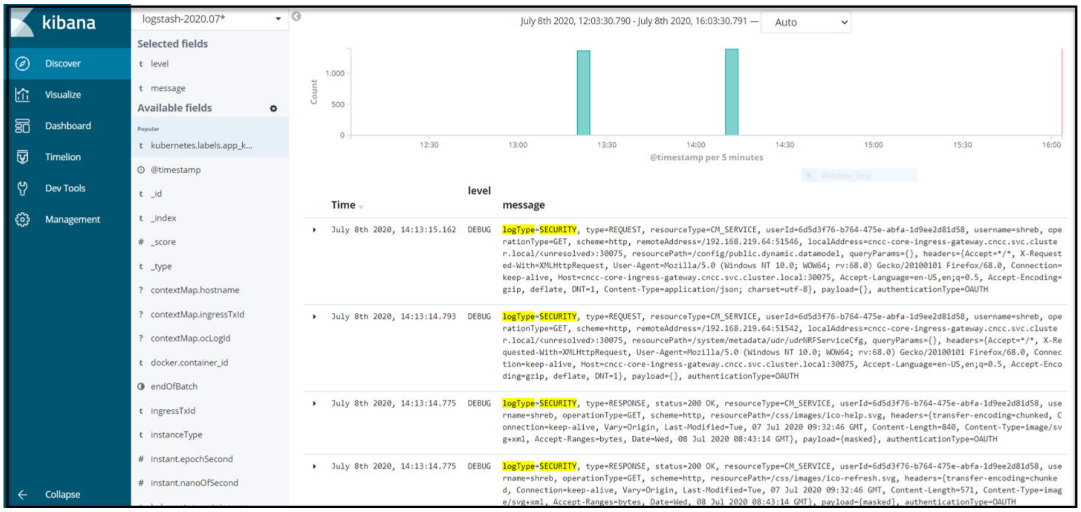
   ```
   $ kubectl logs -f -n cncc cncc-iam-ingress-gateway-77df795fb5-wv2sb -c
   ingress-gateway (Security Log)
   ```

   ```
   $ kubectl logs -f -n cncc cncc-iam-0 (Audit Log)
   ```

2. CNCC uses cloud native supported logging framework to view the logs.

Example : Elasticsearch, Fluentd, and Kibana (EFK) can be used here with CNC Console to view the logs as follows:

Figure 2-1 Log View



```
July 8th 2020, 14:13:15.162  DEBUG  logType=SECURITY, type=REQUEST, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, username=shreb, ope
                                    rationType=GET, scheme=http, remoteAddress=/192.168.219.64:51546, localAddress=cncc-core-ingress-gateway.cncc.svc.cluste
                                    r.local/<unresolved>:30075, resourcePath=/config/public.dynamic.datamodel, queryParams={}, headers={Accept=*/*, X-Request
                                    ed-With=XMLHttpRequest, User-Agent=Mozilla/5.0 (Windows NT 10.0; WOW64; rv:68.0) Gecko/20100101 Firefox/68.0, Connection=
                                    keep-alive, Host=cncc-core-ingress-gateway.cncc.svc.cluster.local:30075, Accept-Language=en-US,en;q=0.5, Accept-Encoding=
                                    gzip, deflate, DNT=1, Content-Type=application/json; charset=utf-8}, payload={}, authenticationType=OAUTH

July 8th 2020, 14:13:14.793  DEBUG  logType=SECURITY, type=REQUEST, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, username=shreb, ope
                                    rationType=GET, scheme=http, remoteAddress=/192.168.219.64:51542, localAddress=cncc-core-ingress-gateway.cncc.svc.cluste
                                    r.local/<unresolved>:30075, resourcePath=/system/metadata/udr/udrNRFServiceCfg, queryParams={}, headers={Accept=*/*, X-Re
                                    quested-With=XMLHttpRequest, User-Agent=Mozilla/5.0 (Windows NT 10.0; WOW64; rv:68.0) Gecko/20100101 Firefox/68.0, Connec
                                    tion=keep-alive, Host=cncc-core-ingress-gateway.cncc.svc.cluster.local:30075, Accept-Language=en-US,en;q=0.5, Accept-Enco
                                    ding=gzip, deflate, DNT=1}, payload={}, authenticationType=OAUTH

July 8th 2020, 14:13:14.775  DEBUG  logType=SECURITY, type=RESPONSE, status=200 OK, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, use
                                    rname=shreb, operationType=GET, scheme=http, resourcePath=/css/images/ico-help.svg, headers={transfer-encoding=chunked, C
                                    onnection=keep-alive, Vary=Origin, Last-Modified=Tue, 07 Jul 2020 09:32:46 GMT, Content-Length=840, Content-Type=image/sv
                                    g+xml, Accept-Ranges=bytes, Date=Wed, 08 Jul 2020 08:43:14 GMT}, payload={masked}, authenticationType=OAUTH

July 8th 2020, 14:13:14.775  DEBUG  logType=SECURITY, type=RESPONSE, status=200 OK, resourceType=CM_SERVICE, userId=6d5d3f76-b764-475e-abfa-1d9ee2d81d58, use
                                    rname=shreb, operationType=GET, scheme=http, resourcePath=/css/images/ico-refresh.svg, headers={transfer-encoding=chunke
                                    d, Connection=keep-alive, Vary=Origin, Last-Modified=Tue, 07 Jul 2020 09:32:46 GMT, Content-Length=571, Content-Type=imag
                                    e/svg+xml, Accept-Ranges=bytes, Date=Wed, 08 Jul 2020 08:43:14 GMT}, payload={masked}, authenticationType=OAUTH
```

2.5.2 Debugging using Logs

This section provides information to debug CNC Console using Logs.

Table 2-3 CNCC Core Debugging through Logs

Scenario	Level	Logs to be searched
CNC Core Login	INFO	Login successful
Session Timeout Value	INFO	Session timeout
Validating user authorization	INFO	User Authorization Details
Accessing a resource	DEBUG	Mapping [Exchange: GET
Updating a resource	DEBUG	Mapping [Exchange: PATCH 'or' Mapping [Exchange: PUT
Creating a new resource	DEBUG	Mapping [Exchange: POST 'or' Mapping [Exchange: PUT
CNCC Core Logout	INFO	Logout successful

3

Debug Tool

Overview

The Debug Tools provides third-party troubleshooting tools for debugging the runtime issues for the lab environment. Following are the available tools:

- tcpdump
- ip
- netstat
- curl
- ping
- dig

> ⓘ **Note**
>
> Debug Tool is only applicable for lab setup, and not recommended for production use.

Prerequisites

This section explains the prerequisites for using debug tool.

- For CNE 23.2.0 and later versions, follow step a of Configuration in CNE.
- For CNE versions prior to 23.2.0, follow step b of Configuration in CNE.

1. **Configuration in CNE**
   The following configurations must be performed in the Bastion Host.
   a. When CNC Console is installed on CNE version 23.2.0 or above

> ⓘ **Note**
>
> - You need admin privileges to edit or patch the clusterpolicies that are mentioned in following steps.
> - In CNE v23.2.0, kyverno policy, disallow-capabilities does not allow NET_ADMIN and NET_RAW capabilities required for debug tool.

Adding Namespace to the Empty Resource

If the current disallow-capabilities cluster policy has no namespace in it, for example:

```
$ kubectl get clusterpolicies disallow-capabilities -oyaml
```

Sample output

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  - exclude:
      resources: {}
```

Then, run the following command to add the namespace:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/resources",
"value": {"namespaces":["<namespace>"]} }]'
```

Example:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/resources",
"value": {"namespaces":["cncc"]} }]'
```

Sample output

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  - exclude:
      resources:
        namespaces:
        - cncc
```

To remove the namespace, run the following:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "replace", "path": "/spec/rules/0/exclude/any/0/
resources", "value": {} }]'
```

Sample output

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
```

```
  - exclude:
      resources: {}
```

Adding a Namespace to the Existing Namespace List

If the current disallow-capabilities cluster policy has already namespaces added in it, for example:

```
$ kubectl get clusterpolicies disallow-capabilities -oyaml
```

Output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  - exclude:
      resources:
        namespaces:
        - namespace1
        - namespace2
        - namespace3
```

Then, run the following command to add your namespace:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/resources/
namespaces/-", "value": "<namespace>" }]'
```

Example:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "add", "path": "/spec/rules/0/exclude/any/0/resources/
namespaces/-", "value": "cncc" }]'
```

Sample Output:

```
 apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  - exclude:
      resources:
        namespaces:
        - namespace1
        - namespace2
        - namespace3
        - cncc
```

To remove the namespace, run the following command:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "remove", "path": "/spec/rules/0/exclude/any/0/resources/
namespaces/<index>"}]'
```

Example:

```
$ kubectl patch clusterpolicy disallow-capabilities --type=json \
  -p='[{"op": "remove", "path": "/spec/rules/0/exclude/any/0/resources/
namespaces/3"}]'
```

Output:

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
...
...
spec:
  rules:
  - exclude:
      resources:
        namespaces:
        - namespace1
        - namespace2
        - namespace3
```

b. **Configurations in CNE Versions Prior to 23.2.0**
The following configurations must be performed in the Bastion Host.

> **Note**
>
> These steps are needed only when you have PSP admission controller enabled in your kubernetes environment.

PodSecurityPolicy (PSP) Creation

i. Log in to the Bastion Host.

ii. Create a new PSP by running the following command. The parameters *readOnlyRootFileSystem*, *allowPrivilegeEscalation*, *allowedCapabilities* are needed by debug container.

> **Note**
>
> Other parameters are mandatory for PSP creation and can be customized as per the CNE environment. Default values are recommended.

PodSecurityPolicy

```
kubectl apply -f - <<EOF
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: debug-tool-psp
spec:
  readOnlyRootFilesystem: false
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - NET_ADMIN
  - NET_RAW
  fsGroup:
    ranges:
    - max: 65535
      min: 1
    rule: MustRunAs
  runAsUser:
    rule: MustRunAsNonRoot
  seLinux:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  volumes:
  - configMap
  - downwardAPI
  - emptyDir
  - persistentVolumeClaim
  - projected
  - secret
EOF
```

Table 3-1 PodSecurityPolicy

Parameter	Description
apiVersion	APIVersion defines the versioned schema of this representation of an object.
kind	Kind is a string value representing the REST resource this object represents.
metadata	Standard object's metadata.
metadata.name	Name must be unique within a namespace.
spec	spec defines the policy enforced.
spec.readOnlyRootFilesystem	Controls whether the containers run with a read-only root filesystem (that is no writable layer).
spec.allowPrivilegeEscalation	Gates whether or not a user is allowed to set the security context of a container to allowPrivilegeEscalation=true.
spec.allowedCapabilities	Provides a list of capabilities that are allowed to be added to a container.
spec.fsGroup	Controls the supplemental group applied to some volumes. RunAsAny allows any fsGroup ID to be specified.
spec.runAsUser	Controls which user ID the containers are run with. RunAsAny allows any runAsUser to be specified.

Table 3-1 (Cont.) PodSecurityPolicy

Parameter	Description
spec.seLinux	RunAsAny allows any seLinuxOptions to be specified.
spec.supplementalGroups	Controls which group IDs containers add. RunAsAny allows any supplementalGroups to be specified.
spec.volumes	Provides a list of allowed volume types. The allowable values correspond to the volume sources that are defined when creating a volume.

Role Creation

Create a role for the PSP by running the following commands:

Role

```
kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: debug-tool-role
  namespace: cncc
rules:
- apiGroups:
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
  resourceNames:
  - debug-tool-psp
EOF
```

Table 3-2 Role

Parameter	Description
apiGroups	APIGroups is the name of the APIGroup that contains the resources.
apiVersion	APIVersion defines the versioned schema of this representation of an object.
kind	Kind is a string value representing the REST resource this object represents.
metadata	Standard object's metadata.
metadata.name	Name must be unique within a namespace.
metadata.namespace	Namespace defines the space within which each name must be unique.
rules	Rules holds all the PolicyRules for this Role
rules.resourceNames	ResourceNames is an optional white list of names that the rule applies to.
rules.resources	Resources is a list of resources this rule applies to.
rules.verbs	Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule.

RoleBinding Creation

Run the following command to attach the service account for your namespace with the role created for the tool PSP:

RoleBinding

```
kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: debug-tool-rolebinding
  namespace: cncc
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: debug-tool-role
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: system:serviceaccounts
EOF
```

Table 3-3 RoleBinding

Parameter	Description
apiVersion	APIVersion defines the versioned schema of this representation of an object.
kind	Kind is a string value representing the REST resource this object represents.
metadata	Standard object's metadata.
metadata.name	Name must be unique within a namespace.
metadata.namespace	Namespace defines the space within which each name must be unique.
roleRef	RoleRef can reference a Role in the current namespace or a ClusterRole in the global namespace.
roleRef.apiGroup	APIGroup is the group for the resource being referenced
roleRef.kind	Kind is the type of resource being referenced
roleRef.name	Name is the name of resource being referenced
subjects	Subjects holds references to the objects the role applies to.
subjects.kind	Kind of object being referenced. Values defined by this API group are "User", "Group", and "ServiceAccount".
subjects.apiGroup	APIGroup holds the API group of the referenced subject.
subjects.name	Name of the object being referenced.

Configuration Changes in CNCC Helm Charts

To enable debug tools container, make changes to *occncc_custom_values_<version>.yaml* file at Global level by setting *extraContainers: ENABLED.*

```
global:
  # Allowed Values: DISABLED, ENABLED
  # Preference is to set  "resources" request and limit to same values to
avoid HPA issues.
```

```
  extraContainers: ENABLED
  debugToolContainerMemoryLimit: 4Gi
  extraContainersVolumesTpl: |
    - name: debug-tools-dir
      emptyDir:
        medium: Memory
        sizeLimit: {{ .Values.global.debugToolContainerMemoryLimit | quote }}
  extraContainersTpl: |
    - command:
        - /bin/sleep
        - infinity
      image: {{ .Values.global.dockerRegistry }}/occncc/ocdebug-
tools:<debugtool_version
      imagePullPolicy: IfNotPresent
      name: {{ printf "%s-tools-%s" (include "getprefix" .) (include
"getsuffix" .) | trunc 63 | trimPrefix "-" | trimSuffix "-"  }}
      resources:
        limits:
          ephemeral-storage: "512Mi"
          cpu: "0.5"
          memory: {{ .Values.global.debugToolContainerMemoryLimit | quote }}
        requests:
          ephemeral-storage: "512Mi"
          cpu: "0.5"
          memory: {{ .Values.global.debugToolContainerMemoryLimit | quote }}
      securityContext:
        allowPrivilegeEscalation: true
        capabilities:
          drop:
          - ALL
          add:
          - NET_RAW
          - NET_ADMIN
        #runAsUser: <user-id>
      volumeMounts:
      - mountPath: /tmp/tools
        name: debug-tools-dir
```

To enable debug tools at service level, make changes to *occncc_custom_values_<version>.yaml* file at service level by setting *extraContainers: USE_GLOBAL_VALUE*

occncc_custom_values_<version>.yaml

```
cncc-iam:
  kc:
    # Allowed Values: DISABLED, ENABLED, USE_GLOBAL_VALUE
    extraContainers: USE_GLOBAL_VALUE

  ingress-gateway:
    # Allowed Values: DISABLED, ENABLED, USE_GLOBAL_VALUE
    extraContainers: USE_GLOBAL_VALUE
```

```
mcncc-core:
  cmservice:
    # Allowed Values: DISABLED, ENABLED, USE_GLOBAL_VALUE
    extraContainers: USE_GLOBAL_VALUE

  ingress-gateway:
    # Allowed Values: DISABLED, ENABLED, USE_GLOBAL_VALUE
    extraContainers: USE_GLOBAL_VALUE




acncc-core:
  ingress-gateway:
    # Allowed Values: DISABLED, ENABLED, USE_GLOBAL_VALUE
    extraContainers: USE_GLOBAL_VALUE
```

Note

Debug Tool Container comes up with the default user ID - 7000. If the operator wants to override this default value, it can be done using the **runAsUser** field, otherwise the field can be skipped.

```
Default value: uid=7000(debugtool) gid=7000(debugtool)
groups=7000(debugtool)
```

To override runAsUser add this line under securityContext in extraContainersTpl

```
runAsUser: <user-id>
```

Configuration Options

Table 3-4 Configuration Options

Parameter	Description
global.extraContainersTpl	Describes the kubernetes container template for the debug-tool.
global.debugToolContainerMemoryLimit	This field describes the memory size (request and limit) and emptyDir volume size for the Debug-tool container.
global.extraContainersTpl.command	String array used for container command.
global.extraContainersTpl.image	Docker image name
global.extraContainersTpl.imagePullPolicy	Image Pull Policy
global.extraContainersTpl.name	Name of the container
global.extraContainersTpl.resources	Compute Resources required by this container
global.extraContainersTpl.resources.limits	Limits describes the maximum amount of compute resources allowed
global.extraContainersTpl.resources.requests	Requests describes the minimum amount of compute resources required
global.extraContainersTpl.resources.limits.cpu	CPU limits

Table 3-4 (Cont.) Configuration Options

Parameter	Description
global.extraContainersTpl.resources.limits.memory	Memory limits
global.extraContainersTpl.resources.limits.ephemeral-storage	Ephemeral Storage limits
global.extraContainersTpl.resources.requests.cpu	CPU requests
global.extraContainersTpl.resources.requests.memory	Memory requests
global.extraContainersTpl.resources.requests.ephemeral-storage	Ephemeral Storage requests
global.extraContainersTpl.securityContext	Security options the container should run with.
global.extraContainersTpl.securityContext.allowPrivilegeEscalation	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This boolen directly controls if the no_new_privs flag will be set on the container process
global.extraContainersTpl.securityContext.capabilities	The capabilities to add or drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
global.extraContainersTpl.securityContext.capabilities.drop	Removed capabilities
global.extraContainersTpl.secuirtyContext.capabilities.add	Added capabilities
global.extraContainersTpl.securityContext.runAsUser	The UID to run the entrypoint of the container process.
global.extraContainersTpl.volumeMounts.mountPath	The path where the emptyDir volume has to be mounted inside the container.
global.extraContainersTpl.volumeMounts.name	Name of the emptyDir volume.

Debug Tool Volume Parameters

Table 3-5 Debug Tool Volume Parameters

Parameter	Description
global.extraContainersVolumesTpl	Describes the kubernetes volume template for the debug-tool container
global.extraContainersVolumesTpl.name	Name of the emptyDir volume.
global.extraContainersVolumesTpl.emptyDir.sizeLimit	The size of the emptyDir volume.
global.extraContainersVolumesTpl.emptyDir.medium	Describes where emptyDir volumes are stored.

Debug Tool Usage

Following is the procedure to run Debug Tool:

1. Run the following command to retrieve the POD details:

   ```
   $ kubectl get pods -n <k8s namespace>
   ```

   Example:

   ```
   # kubectl get pod -n cncc
   ```

   Sample Output:

   ```
    NAME                                       READY   STATUS    RESTARTS
   AGE
   cncc-acore-ingress-gateway-764f7f5f77-qnr5p   2/2     Running    0
   19m
   cncc-iam-ingress-gateway-55987f7dc9-x5nt2     2/2     Running    0
   147m
   cncc-iam-kc-0                                 2/2     Running    0
   147m
   cncc-mcore-cmservice-947cf4c89-76vq6          2/2     Running    0
   19m
   cncc-mcore-ingress-gateway-764f7f5f77-qnr5p   2/2     Running    0
   19m
   ```

2. Run the following command to enter Debug Tools Container:

   ```
   kubectl exec -it <pod name> -c <debug_container name> -n <namespace> bash
   ```

   Example:

   ```
   kubectl exec -it cncc-mcore-ingress-gateway-599d858867-x9pvz -c tools -n
   cncc bash
   ```

3. Run the debug tools:

   ```
   bash -4.2$ <debug_tools>
   ```

   Example:

   ```
   bash -4.2$ tcpdump
   ```

4. Run the following command to copy output files from container to host:

   ```
   $ kubectl cp -c <debug_container name>  <pod name>:<file location in
   container> -n <namespace>  <destination location>
   ```

   Example:

   ```
   $ kubectl cp -c tools -n cncc cncc-mcore-ingress-gateway-764f7f5f77-
   qnr5p:/tmp/capture.pcap /tmp/
   ```

Steps to Enable Debug Tools Container

Debug tools container can be enabled or disabled for CNCC components by using helm install or helm upgrade command.

CNC Console

Run the following command to enable or disable CNC Console IAM after updating *occncc_custom_values_<version>.yaml* file on a installed setup:

```
$ helm upgrade <release_name> -f occncc_custom_values_<version>.yaml <helm-
repo> --version <helm_version>
```

Example:

```
$ helm upgrade cncc -f occncc_custom_values_23.4.2.yaml ocspf_helm-repo/cncc
--version 23.4.2
```

4

CNC Console Troubleshooting

This section provides information to troubleshoot the common errors which can be encountered during the installation and upgrade of CNC Console.

Unable to display the release version of the NF at CNC Console banner

Problem: CNC Console banner displays the release version of CNC Console, but not displaying the release version of the NF.

Solution:

- The “About” section and Application name displayed next to Oracle logo use the envSystemName and envNFVersion helm fields.
- The value set of *envSystemName* and *envNFVersion* combines to display the Application name (Application name = envSystemName + envNFVersion).
- CNC Console Core Custom values have *envSystemName* and *envNFVersion* mentioned in it, but these values can be overridden.

Unable to reach CNC Console Core IP or port directly

Problem: Unable to reach CNC Console Core IP or port directly. *redirect_uri* is inserted instead of directly accessing the CNC Console Core.

Solution: As per the design, CNC Console redirects requests to CNC Console IAM for authentication. On successful authentication, CNC Console IAM redirects the user back to CNC Console GUI.

'Admin' user created under CNC Console realm is unable to access CNC Console IAM

Problem: The user with ‘Admin’ privileges is unable to access CNC Console IAM.

Solution: Users created under the *Cncc* realm have access only to CNC Console Core and not to CNC Console IAM. To access CNC Console IAM, create the admin user under the *Master* realm.

CNC Console returns 403 error during NF Configuration

Problem: CNCConsole returns a 403 Error Code and error *"Forbidden. Data could not be saved".*

Error Code/Error Message:
403/Forbidden

Solution: Log into CNC Console IAM to check the roles of the user. The user must have <NF>_READ and <NF>_WRITE roles assigned to perform the write operation on any NF through the CNC Console.

CNC Console returns 500 - Internal Server Error

Problem: CNC Console returns a 500 Error Code while accessing NF Resource.

Error Code/Error Message:
500/Internal Server Error

Solution: The internal server error occurs when the NF routes are not configured correctly. To resolve this error, ensure that correct routes for each NF are configured during deployment. You can provide routes in either of the IP/FQDN in the Instances section:

```
id: <Instance ID>
type: <NF type>
owner: <ID of cluster owning the Instance>
ip: <IP of NF deployment>
port: <Port of NF deployment
```

CNC Console IAM is accessible, but CNC Console Core is not accessible

Problem: CNC Console IAM is accessible, but CNC Console Core is not accessible.

Error Message:
The ID Token contains invalid claims, which is a JWT validation error, indicating that the system clock on your server is off.

Observation: This issue occurs when Ingress Gateway is behind in time and when CNC Console IAM is ahead of time. For example, If IAM (node1) is ahead of time and Ingress Gateway (node2) is 5 minutes behind, the Ingress Gateway invalidates the received token and throws *"The ID Token contains invalid claims: {iat=2020-05-26T08:32:12Z}"* error.

Solution: To resolve the error, you must ensure that the same time is maintained in CNC Console IAM and Ingress Gateway when they run in the same instance or different NTP server instances.

CNC Console IAM admin password configured through Kubectl secret is not reflected

Problem:
CNC Console IAM admin password change through cncc-iam-secret is not working (Example: if configured cncc-iam-secret).

Solution: During the first installation, CNC Console IAM reads the password from the cncc-iam-secret and stores it in the database. So any further changes to the admin password must be done through the CNC Console IAM GUI.

Access Error in CNC Console Core GUI

Problem:

Unable to access CNC Console Core GUI and an “Invalid redirect URI” error occurs.

Observation:

This error occurs when there is a mismatch between the Root URL provided in CNC Console IAM Admin Console and the URI through which you access the CNC Console Core GUI.

For example, In CNC Console IAM, the Root URL is mentioned as *http://cncc-core-ingress-gateway.cncc.svc.cluster.local:30075/* and if you are accessing the CNC Console Core GUI with IP and NodePort, that is, *http://10.75.xx.xx:30075/** or vice-versa, you get “invalid redirect_uri” error on CNC Console Core GUI.

Solution: To resolve this error, ensure that the Root URL provided in CNC Console IAM and the URI through which you access the CNC Console Core GUI are the same.

Changing the CNC Console IAM admin password

Problem:
How to change the CNC Console IAM admin password using the REST API call.

Solution: Refer the following sections in CNC Console User Guide:

- Accessing NF Resources through Curl or Postman
- CNC Console IAM REST APIs

Unable to access Kibana

Problem:
Kibana Common Service is not accessible

Solution: To resolve this issue, ensure that you are accessing Kibana through the correct path. The default access path to Kibana is through "/kibana". You can also access Kibana through the URL *<node-ip>:<node-port>/mycne-cluster/kibana.*

CNC Console Installation failure while installing using cnDBTier

Problem:
While installing CNC Console using cnDBTier, the cncc-iam-kc pod does not come up and goes into a crash state.

Solution: cnDBTier needs additional grants such as "REFERENCES, INDEX" due to the addition of db hook job.

CNC Console IAM kc pod fails while ASM is enabled

Problem:
While ASM is enabled, CNC Console IAM kc pod fails due to Readiness probe failure.

Solution: Check whether annotation *"sidecar.istio.io/rewriteAppHTTPProbers"* is enabled and set to true under *'nonlbStatefulSets'* in *custom_cncc-iam_values.yaml* during CNC Console IAM deployment.

Unable to Access CNC Console GUI when ASM is Enabled

Problem:
Unable to access CNC Console GUI after installation as cncc-iam-ingress-gateway is listening on port 8080 instead of port 8081(ASM enabled).

Solution: After installing CNC Console, the cncc-iam-ingress-gateway is listening on port 8080 instead of port 8081 when ASM is enabled. To resolve this issue, configure the parameters in the *custom_cncc-iam_values.yaml* file as follows:

- Annotation: sidecar.istio.io/rewriteAppHTTPProbers: "\"true\""
- serviceMeshCheck: true
- Annotation: sidecar.istio.io/inject: "true"

CNC Console Core GUI does not get loaded after logging in

Problem
CNC Console Core microservices are up and running but CNC Console Core GUI does not get loaded after logging in.

Solution
CNC Console supports only single pod deployment, check the following configurations (must be set to 1).

```
ingress-gateway:
  # Number of Pods must always be available, even during a disruption.
  minAvailable: 1
  # Min replicas to scale to maintain an average CPU utilization
  minReplicas: 1
  # Max replicas to scale to maintain an average CPU utilization
  maxReplicas: 1
```

> **Note**
>
> These are preset to 1 and these parameters are not exposed in custom values.

CNC Console is not supporting ASM with mTLS disabled configuration

Problem

When service mesh is enabled and mTLS is set either to disabled or to permissive with insecure HTTP connections, CNC Console Core microservice doesn't come up or CNC Console Core GUI takes time to load because some internal CSS ir JS calls are still triggered with HTTPs.

Solution

1. Update *serviceMeshHttpsEnabled* to false in *custom-cncc-core_values.yaml* file to allow insecure HTTP connections.

   ```
   #Mandatory: This parameter must be set to "true" when CNC Console is
   deployed with the Service Mesh
   serviceMeshCheck: true
   # If Service Mesh is deployed with TLS/MTLS disabled then set this flag to
   false
   serviceMeshHttpsEnabled: false
   ```

2. Add a PeerAuthentication rule stating mTLS is disabled for CNC Console namespace. For example,

   ```
   kubectl apply -n <namespace> -f - <<EOF
   apiVersion: security.istio.io/v1beta1
   kind: PeerAuthentication
   metadata:
     name: cnccpeer
   spec:
     mtls:
       mode: DISABLE
   EOF

   ##################################
   # Example with cncc as namespace #
   ##################################
   kubectl apply -n cncc -f - <<EOF
   apiVersion: security.istio.io/v1beta1
   kind: PeerAuthentication
   metadata:
     name: cnccpeer
   spec:
     mtls:
       mode: DISABLE
   EOF
   ```

Pods not coming up in an ASM enabled CNC Console deployment in istio injected namespace

Problem

CNC Console pods don't come up when sidecar injection is enabled and CNC Console is deployed in CNE, and the following error is seen:

```
job/cncc-iam-pre-install          Error creating: pods "cncc-iam-pre-
install-" is forbidden: PodSecurityPolicy: unable to admit pod:
 [spec.initContainers[0].securityContext.runAsNonRoot: Invalid value: false:
must be true spec.initContainers[0].securityContext: Invalid value:
[]int64{0}: group 0 must be in the ranges: [{1 65535}]
spec.initContainers[0].securityContext.capabilities.add: Invalid value:
"NET_ADMIN": capability may not be added
spec.initContainers[0].securityContext.capabilities.add: Invalid value:
"NET_RAW": capability may not be added
spec.initContainers[0].securityContext.runAsNonRoot: Invalid value: false:
must be true]
```

Solution

You must bind the ClusterRole "psp:privileged" to the current namespace:

```
kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: "psp:<namespace>:cs-restricted"
  namespace: "<namespace>"
roleRef:
  kind: ClusterRole
  apiGroup: rbac.authorization.k8s.io
  name: "psp:privileged"
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: "system:serviceaccounts"
EOF
```

Failed to allocate IP for CNC Console IAM Ingress Gateway

Problem

Installation of CNC Console IAM is successful but while checking CNC Console IAM service status, unable to assign the external IP for svc cncc-iam-ingress-gateway and received the following error: *Warning Allocation Failed 61s (x3 over 8m48s) metallb-controller Failed to allocate IP for "cncc/cncc-iam-ingress-gateway": no available IPs.*

Solution

Check if the annotations are missing from the cncc-iam-ingress-gateway service. Add the missing annotations, due to which the dynamic metalLbIpAllocation will work properly.

Unable to Create required tables in CNC Console IAM DB

Problem

Deployment needs two instances of CNC Console where only the first instance is deployed correctly. After installing the second instance of CNC Console in a different namespace, the pod "cncc-voice-iam-kc-0" repeatedly crashes

Observation

After analyzing the logs, it was found that during the preinstall checks, the hook pods did not create all the required tables in the DB. For example, in the first instance DB, all tables created, while in the second instance DB, there are only 43 tables created. The cbDBTier has a maximum table limit of 512. So, during the deployment of the second instance of CNC Console, the maximum table limit threshold has exceeded, and hence 43 tables were created.

Deployment needs two instances of CNC Console, first instance is deployed correctly.

After the installation of the second cncc-iam in a different namespace, the *pod cncc-voice-iam-kc-0* is crashed repeatedly. By analyzing the logs it seems that not all the tables has been created by the hook pods during the pre-install checks. In the first instance db we can see that there are all tables created while in the second instance we can see only 43 tables.

Solution
To resolve this issue, you must either increase the maximum table limit or clean up unwanted databases to bring table count within the threshold limits. For more information about configuring the table limits, see *Oracle Communications Cloud Native Core DBTier Installation and Upgrade Guide*.

Default limits in ndb:

MaxNoOfOrderedIndexes: 512

MaxNoOfTables: 512

NoOfFragmentLogFiles: 256

CNC Console Core Installation Failing with Validation Hook Error ProblemCNCC Core installation is failing with the validation hook error.SolutionTo resolve this error, in the custom-cncc-core_values.yaml file, check if the multiClusterMultiInstanceEnabled parameter is set to true. When this flag is enabled, the preinstall hook "cncc-core-validation-hook" starts validating the multicluster deployment configurations.If you do not need CNC Console multicluster deployment validation, then set the multiClusterMultiInstanceEnabled parameter to false.

Resolve CNC Console Validation hook error

Problem

Validation hook error occurs during CNC Console Core Deployment.

Solution

To resolve this issue, enable Helm Configuration Validation for CNC Console Deployment, applicable for M-CNCC Core and A-CNCC deployment.

Check the *cncc-acore-validation-hook* or *cncc-mcore-validation-hook* pod logs for the error codes. Make the required corrections in the *custom-cncc-core_values.yaml* file and reinstall M-CNCC Core or A-CNCC. For more information about validation hook and error details, see "CNC Console Multi Cluster Deployment Helm Configuration Validation" section in *Oracle Communications Cloud Native Configuration Console Installation and Upgrade Guide.*

Error Code	Error Message Format	Error Scenarios	Sample Error Messages
1001	Invalid value. Resource: <Configuration Name>, ID: <ID>, Attribute: <Attribute>. <More Info>	• Port should be Numeric • Scheme should be either HTTP/HTTPS • IDs should follow the alphanumeric pattern • Max Limit should be satisfied for M-CNCC IAM, A-CNCC and Instance • Max Length for Instance Id • Max Length for Self Cncc Id • CS instance must have one of these CS subtypes <grafana, kibana, jaeger, prometheus, alertmanager> • Both ip and fqdn cannot be provided. • Unsupported type • InvalidConfig • multicluster flag should be false in case of single-cluster deployment • multicluster flag should be true in case of multi-cluster deployment	Invalid value. Resource: mCnccIam, ID: Cluster1, Attribute: Port. It should be numeric value. Invalid value. Resource: instance, ID: Cluster3 Cluster3-instance1, Attribute: Scheme. Allowed values are: [http, https]. Invalid value. Resource: instance, ID: Cluster1-grafana##$$%, Attribute: id. Ids should be alphanumeric with hyphen allowed as special character. The count of mCnccIam exceeded max limit. Allowed Value:x. Actual Value: y Max limit exceeded. Allowed Value:x. Actual Value: y Invalid value. Resource: aCncc, ID: Cluster3, Attribute: N/A. Both ip and fqdn cannot be provided. Invalid value. Resource: isMultiClusterEnabled, ID:,Attribute: False. isMultiClusterEnabled is set as false, only single cluster configuration is allowed. Invalid value. Resource: isMultiClusterEnabled, ID:,Attribute: True. isMultiClusterEnabled is set as true, only multi cluster configuration is allowed.
1002	Duplicate value. Resource: <Configuration Name>, ID: <ID>, Attribute: <Attribute>. <More Info>	• All A-CNCC IDs must be unique • API prefix must be unique for all instances • Owner(Cluster) must have unique CS subtype	Duplicate value(s). Resource: aCncc, ID: [Cluster3], Attribute: id.
1003	Invalid Reference. Resource: <Configuration Name>, ID: <ID>, Attribute: <Attribute>. <More Info>	• All the Instance owners must be referenced in M-CNCC IAM IDs or A-CNCC IDs • M-CNCC IAM IDs and M-CNCC Core IDs must be same	Invalid Reference. Resource: instance, ID: Cluster5, Attribute: Owner. Not present in mCncc ids or aCncc ids. Invalid Reference. Resource: instance, ID: N/A, Attribute: N/A. M-Cncc Iam ids and M-Cncc Core ids do not match.

Error Code	Error Message Format	Error Scenarios	Sample Error Messages
1004	Missing value. Resource: <Configuration Name>, ID: <ID>, Attribute: <Attribute>. <More Info>	• Missing apiPrefix parameter for type CS • Either of IP/FQDN should be present	Missing value. Resource: instance, ID: Cluster4-grafana, Attribute: apiPrefix. Missing value. Resource: instance, ID: Cluster3-PolicyInstance, Attribute: N/A. Either ip or fqdn is required.

How to Reset CNC Console IAM Admin Password

Problem: How to Reset CNC Console IAM Admin Password

Solution: To reset the CNC Console IAM admin password, perform the following procedure:

1. Run the following command to get into the mysql pod of cnDBTier:

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

2. Run the following command to login to mysql:

   ```
   mysql -h 127.0.0.1 -u <username> -p<password>
   ```

3. Run the following command to use the database that you used to deploy the CNC Console application:

   ```
   use <database name>
   ```

4. Run the following command to find the admin user ID from the USER_ENTITY table:

   ```
   select * from USER_ENTITY
   ```

5. Run the following command to delete the admin user ID entry from CREDENTIAL table:

   ```
   delete from CREDENTIAL where user_id = '<user-id>'
   ```

6. Run the following command to delete the admin user ID entry from ROLE_MAPPING table:

   ```
   delete from USER_ROLE_MAPPING where user_id = '<user-id>'
   ```

7. Run the following command to delete the admin user ID entry from USER_ENTITY table:

   ```
   delete from USER_ENTITY where ID = '<user-id>'
   ```

8. Run the following command to restart the IAM-KC-0 pod and login to CNC Console IAM using the password given in the cncc-iam secret:

   ```
   Kubectl delete po <IAM-KC Pod> -n <namespace>
   ```

Does CNC Console support Command Line Interface (CLI)

Problem: Can NF APIs integrated with CNC Console be accessed through curl or postman.

Solution The NF configuration APIs can be accessed through CNC Console GUI or directly using postman or curl. CNC Console providess authentication and authorization in both ways. For more information, see "Generating Access Tokens and Accessing NF Resources" section in *Oracle Communications Cloud Native Configuration Console User Guide.*

Upgrade or Rollback Failure

Problem: Upgrade or Rollback Failure

Solution

When CNC Console upgrade or rollback fails, perform the following procedure:

1. Check the pre or post upgrade or rollback hook logs as applicable.
2. If the failure occurs, then check the cause of the failure from the logs by running the following command:

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

3. After detecting the cause of failure, do the following:
   - For upgrade failure:
     - If the cause of upgrade failure is database or network connectivity issue, then resolve the issue and rerun the upgrade command.
     - If the upgrade failure occurs during the postupgrade phase, for example, post upgrade hook failure due to target release pod not moving to ready state, then perform a rollback.

- For rollback failure: If the cause of rollback failure is database or network connectivity issue, then resolve the issue and rerun the rollback command.

4. If the issue persists, contact My Oracle Support.

CNC Console Upgrade Results IP in Pending state

Problem: CNC Console deployment using static IP is not allocated to the new mcore service during upgrade.

Solution

CNCC supports the single helm chart deployment for deploying all three components M-CNCC IAM, M-CNCC Core and A-CNCC Core.

Earlier CNCC IAM and CNCC Core were deployed independently, now with single helm chart all 3 components can be deployed using single helm install command.

Upgrade from two helm deployments to one helm deployment is supported but one of the helm deployment must be manually deleted.

CNCC IAM deployment can be upgraded which upgrades M-CNCC IAM and freshly install M-CNCC Core and A-CNCC Core services. User can manuallydelete CNCC Core deployment. For more information, see Upgrade and Rollback sections of *Oracle Communications Cloud Native Configuration Console Installation and Upgrade Guide.*.

In case, if static LoadBalancer IP is used in existing deployment, after the upgrade, new mcore service IP will be shown as pending. IP will be allocated once the existing M-CNCC Core service is uninstalled.

CNC Console Upgrade Displays Port Already in Use Error

Problem: CNC Console deployment using static node port throws P*ort already in use error* during upgrade.

Solution

If static port is used in existing deployment, before upgrade, in custom values file port needs to be updated to use another port to avoid port conflict error.

CNC Console Helm Test Fails

Problem: CNC Console helm test fails when there are stale jobs or pods.

Solution

In some cases, Helm RC builds have intermittent issues which blocks auto deletion of jobs.

Ensure stable helm version is installed in your environment.

CNC Console Helm Test Fails with Service Account Error

Problem: CNC Console helm test fails when there are stale jobs or pods.

CNCC helm test fails with error message "Unauthorized! Configured service account doesn't have access. Service account may have been revoked.".

Solution

The time sync between worker nodes is must for helm test to work. Ensure CNE worker nodes time is in sync.

5

CNC Console Alerts

This section provides information about CNC Console Alerts.

Note

Alert file is present in the Scripts directory. occncc_csar_<version>.zip can be downloaded from MOS. Unzip the file to get occncc_alertrules_<version>.yaml file.

- Review the o*occncc_alerting_rules_promha_<version>.yaml* file and edit the value of the parameters in the *occncc_alerting_rules_promha_<version>.yaml* file (if needed to be changed from default values) before configuring the alerts.
- *kubernetes_namespace* is configured as kubernetes namespace in which CNCC is deployed. Default value is cncc. Update the *occncc_alertrules_<version>.yaml* file to reflect the correct CNCC kubernetes namespace.

Two sample Alert files are provided, one for supporting CNE 1.8 or lower and second one supporting CNE Prometheus HA.

- CNCC Alert Rules file: *occncc_alertrules_<version>.yaml* file.
- CNCC Alert Rules file supporting CNE Prometheus HA: *occncc_alerting_rules_promha_<version>.yaml* file.

5.1 CNC Console IAM Alerts

This section provides information about CNC Console IAM Alerts.

5.1.1 CnccIamTotalIngressTrafficRateAboveMinorThreshold

Table 5-1 CnccIamTotalIngressTrafficRateAboveMinorThreshold

Trigger Condition	The total CNCC IAM Ingress Message rate has crossed the configured minor threshold of 700 TPS. Default value of this alert trigger point in *occncc_alertrules_<version>.yaml*is when CNCC IAM Ingress Rate crosses 70 % of 1000 (Maximum ingress request rate)
Severity	Minor

Table 5-1 (Cont.) CncclamTotalIngressTrafficRateAboveMinorThreshold

Alert details provided	**Description**: CNCC IAM Ingress traffic Rate is above the configured minor threshold i.e. 700 requests per second (current value is: {{ $value }}) **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Traffic Rate is above 70 Percent of Max requests per second(1000)' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is above 70 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7001
Metric Used	oc_ingressgateway_http_requests_total
Resolution	The alert is cleared either when the total Ingress Traffic rate falls below the Minor threshold or when the total traffic rate crosses the Major threshold, in which case the *CncclamTotalIngressTrafficRateAboveMajorThreshold* alert is raised. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. Steps: 1. Reassess why the CNCC IAM is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

5.1.2 CncclamTotalIngressTrafficRateAboveMajorThreshold

Table 5-2 CncclamTotalIngressTrafficRateAboveMajorThreshold

Trigger Condition	The total CNCC IAM Ingress Message rate has crossed the configured major threshold of 800 TPS. Default value of this alert trigger point ino *ccncc_alertrules_<version>.yaml* is when CNCC IAM Ingress Rate crosses 80 % of 1000 (Maximum ingress request rate)
Severity	Major
Alert details provided	**Description**: 'CNCC IAM Ingress traffic Rate is above the configured major threshold i.e. 800 requests per second (current value is: {{ $value }})' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7001

Table 5-2 (Cont.) CncclamTotalIngressTrafficRateAboveMajorThreshold

Metric Used	oc_ingressgateway_http_requests_total
Resolution	The alert is cleared when the total Ingress Traffic rate falls below the Major threshold or when the total traffic rate crosses the Critical threshold, in which case the *CncclamTotalIngressTrafficRateAboveCriticalThreshold* alert is raised. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. Steps: 1. Reassess why the CNCC IAM is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

5.1.3 CncclamTotalIngressTrafficRateAboveCriticalThreshold

Table 5-3 CncclamTotalIngressTrafficRateAboveCriticalThreshold

Trigger Condition	The total CNCC IAM Ingress Message rate has crossed the configured critical threshold of 900TPS. Default value of this alert trigger point in *occncc_alertrules_<version>.yaml* is when CNCC IAM Ingress Rate crosses 90 % of 1000 (Maximum ingress request rate)
Severity	Critical
Alert details provided	**Description**:CNCC IAM Ingress traffic Rate is above the configured critical threshold, that is, 900 requests per second (current value is: {{ $value }}) **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7001
Metric Used	oc_ingressgateway_http_requests_total
Resolution	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. Steps: 1. Reassess why the CNCC IAM is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

5.1.4 CnccIamMemoryUsageCrossedMinorThreshold

Table 5-4   CnccIamMemoryUsageCrossedMinorThreshold

Trigger Condition	A pod has reached the configured minor threshold(70%) of its memory resource limits.
Severity	Minor
Alert details provided	**Description**: 'CNCC IAM Memory Usage for pod {{ $labels.pod }} has crossed the configured minor threshold (70%) (value={{ $value }}) of its limit.' **Summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 70% of its limit.'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7002
Metric Used	container_memory_usage_bytes, kube_pod_container_resource_limits **Note:**This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use a similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the memory utilization falls below the Minor Threshold or crosses the major threshold, in which case *CnccIamMemoryUsageCrossedMajorThreshold* alert is raised. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. If guidance is required, contact My Oracle Support.

5.1.5 CnccIamMemoryUsageCrossedMajorThreshold

Table 5-5   CnccIamMemoryUsageCrossedMajorThreshold

Trigger Condition	A pod has reached the configured major threshold(80%) of its memory resource limits.
Severity	Major
Alert details provided	**Description**: 'CNCC IAM Memory Usage for pod {{ $labels.pod }} has crossed the configured major threshold (80%) (value = {{ $value }}) of its limit.' **Summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 80% of its limit.'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7002
Metric Used	container_memory_usage_bytes, kube_pod_container_resource_limits **Note:** This is a Kubernetes metric used for instance availability monitoring.If the metric is not available, use a similar metric as exposed by the monitoring system.

Table 5-5 (Cont.) CncclamMemoryUsageCrossedMajorThreshold

Resolution	The alert gets cleared when the memory utilization falls below the Major Threshold or crosses the critical threshold, in which case *CncclamMemoryUsageCrossedCriticalThreshold* alert shall be raised. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. If guidance is required, contact My Oracle Support.

5.1.6 CncclamMemoryUsageCrossedCriticalThreshold

Table 5-6 CncclamMemoryUsageCrossedCriticalThreshold

Trigger Condition	A pod has reached the configured critical threshold (90%) of its memory resource limits
Severity	Critical
Alert details provided	**Description**: 'CNCC IAM Memory Usage for pod {{ $labels.pod }} has crossed the configured critical threshold (90%) (value = {{ $value }}) of its limit.' **Summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 90% of its limit.'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7002
Metric Used	container_memory_usage_bytes, kube_pod_container_resource_limits **Note:** This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use a similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the memory utilization falls below the Critical Threshold. Note : The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. If guidance is required, contact My Oracle Support.

5.1.7 CncclamTransactionErrorRateAbove0.1Percent

Table 5-7 CncclamTransactionErrorRateAbove0.1Percent

Trigger Condition	The number of failed transactions is above 0.1 percent of the total transactions.
Severity	Warning
Alert details provided	**Description**: 'CNCC IAM transaction Error rate is above 0.1 Percent of Total Transactions (current value is {{ $value }})' **Summary**: 'CNCC IAM transaction Error Rate detected above 0.1 Percent of Total Transactions'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7003
Metric Used	oc_ingressgateway_http_responses_total

Table 5-7 (Cont.) CncclamTransactionErrorRateAbove0.1Percent

Resolution	The alert is cleared when the number of failed transactions is below 0.1% of the total transactions or when the number of failed transactions crosses the 1% threshold in which case the *CncclamTransactionErrorRateAbove1Percent* is raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.1.8 CncclamTransactionErrorRateAbove1Percent

Table 5-8 CncclamTransactionErrorRateAbove1Percent

Trigger Condition	The number of failed transactions is above 1 percent of the total transactions.
Severity	Warning
Alert details provided	**Description**: 'CNCC IAM transaction Error rate is above 1 Percent of Total Transactions (current value is {{ $value }})' **Summary**: 'CNCC IAM transaction Error Rate detected above 1 Percent of Total Transactions'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7003
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert is cleared when the number of failed transactions is below 1% of the total transactions or when the number of failed transactions crosses the 10% threshold in which case the CncclamTransactionErrorRateAbove10Percent is raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.1.9 CncclamTransactionErrorRateAbove10Percent

Table 5-9 CncclamTransactionErrorRateAbove10Percent

Trigger Condition	The number of failed transactions is above 10 percent of the total transactions.
Severity	Minor
Alert details provided	**Description**: CNCC IAM transaction Error rate is above 10 Percent of Total Transactions (current value is {{ $value }})' **Summary**: 'CNCC IAM transaction Error rate is above 10 Percent of Total Transactions (current value is {{ $value }})'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7003
Metric Used	oc_ingressgateway_http_responses_total

Table 5-9 (Cont.) CncclamTransactionErrorRateAbove10Percent

Resolution	The alert is cleared when the number of failed transactions is below 10% of the total transactions or when the number of failed transactions crosses the 25% threshold in which case the *CncclamTransactionErrorRateAbove25Percent* is raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.1.10 CncclamTransactionErrorRateAbove25Percent

Table 5-10 CncclamTransactionErrorRateAbove25Percent

Trigger Condition	The number of failed transactions is above 25 percent of the total transactions.
Severity	Major
Alert details provided	**Description**: 'CNCC IAM transaction Error Rate detected above 25 Percent of Total Transactions (current value is {{ $value }})' **Summary**: 'CNCC IAM transaction Error Rate detected above 25 Percent of Total Transactions'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7003
Metric Used	oc_ingressgateway_http_responses_total
Resolution	TThe alert is cleared when the number of failed transactions are below 25% of the total transactions or when the number of failed transactions cross the 50% threshold in which case the *CncclamTransactionErrorRateAbove50Percent* is raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.1.11 CncclamTransactionErrorRateAbove50Percent

Table 5-11 CncclamTransactionErrorRateAbove50Percent

Trigger Condition	The number of failed transactions is above 50 percent of the total transactions.
Severity	Critical
Alert details provided	**Description**: The number of failed transactions is above 50 percent of the total transactions. **Summary**: 'CNCC IAM transaction Error Rate detected above 50 Percent of Total Transactions'.
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7003
Metric Used	oc_ingressgateway_http_responses_total

Table 5-11 (Cont.) CncclamTransactionErrorRateAbove50Percent

Resolution	The alert is cleared when the number of failed transactions is below 50 percent of the total transactions. The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.1.12 CncclamIngressGatewayServiceDown

Table 5-12 CncclamIngressGatewayServiceDown

Trigger Condition	The pods of the CNCC IAM Ingress Gateway microservice is available.
Severity	Critical
Alert details provided	**Description**:'CNCC IAM Ingress-Gateway service InstanceIdentifier=~".*cncc-iam_ingressgateway" is down' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Ingress-gateway service down' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Ingress-gateway service down'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7004
Metric Used	'up' **Note:** This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.
Resolution	The alert is cleared when the cncc-iam_ingressgateway service is available. Steps: 1. Check the orchestration logs of cncc-iam_ingressgateway service and check for liveness or readiness probe failures. 2. Refer to the application logs on Kibana and filter based on cncc-iam_ingressgateway service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.

5.1.13 CncclamFailedLogin

Table 5-13 CncclamFailedLogin

Trigger Condition	The count of failed login attempts in CNCC-IAM by a user goes above '3'
Severity	Warning
Alert details provided	**Description**:'{{ $value }} failed Login attempts have been detected in CNCC IAM for user {{$labels.UserName}}, the configured threshold value is 3 failed login attempts for every 5 min' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: failed login attempts are more than the configured threshold value' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: failed login attempts are more than the configured threshold value'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7005
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert gets cleared when the total failed login attempts for a particular user goes below the threshold value (default value is '3') in the last 5 min (default value is 5 m). **Note:** The threshold and time is configurable in the *alerts.yaml* file. If guidance is required, contact My Oracle Support.

5.1.14 AdminUserCreation

Table 5-14 AdminUserCreation

Trigger Condition	If a new admin account is created in the last 5 min
Severity	Warning
Alert details provided	**For CNE 1.9.0 or later versions:** **Description**: '{{ $value }} admin users have been created by {{$labels.UserName}} ' summary: 'namespace: {{$labels.namespace}} **summary**: {{$labels.pod}}, user: {{$labels.UserName}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Admin users have been created ' **For CNE 1.8.x or previous versions:** **Description**: '{{ $value }} admin users have been created by {{$labels.UserName}} ' summary: 'namespace: {{$labels.kubernetes_namespace}} **summary**: {{$labels.kubernetes_pod_name}}, user: {{$labels.UserName}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Admin users have been created '
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7006
Metric Used	oc_ingressgateway_http_requests_total

Table 5-14 (Cont.) AdminUserCreation

Resolution	The alert gets cleared when the total failed login attempts for a particular user go below the threshold value (default value is '3') in the last 5 min (default value is 5 m) **Note:** The threshold and time is configurable in the *occncc_alertrules_<version>.yaml* file. Login to admin GUI and review the user created. If guidance is required, contact My Oracle Support.

5.1.15 CncclamAccessTokenFailure

Table 5-15 CncclamAccessTokenFailure

Trigger Condition	If the count of failed token for CNCC-IAM goes above configured value of '3'
Severity	Warning
Alert details provided	**Description**: 'CNCC Iam Access Token Failure count is above the configured value i.e. 3 for every 5 min. Failed access token request count per second is (current value is: {{ $value }})' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Access Token Failure count is above the configured threshold value' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Access Token Failure count is above the configured threshold value'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.7007
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert gets cleared when the total failed tokens for a particular user go below the threshold value (default value is '3') in the last 5 min (default value is 5 m) **Note:** The threshold and time is configurable in the *occncc_alertrules_<version>.yaml* file. If guidance is required, contact My Oracle Support.

5.2 CNC Console Core Alerts

This section provides the information about CNC Console Core Alerts.

5.2.1 CnccCoreTotalIngressTrafficRateAboveMinorThreshold

Table 5-16 CnccCoreTotalIngressTrafficRateAboveMinorThreshold

Trigger Condition	The total CNCC Core Ingress Message rate has crossed the configured minor threshold of 700 TPS. Default value of this alert trigger point in cncc_alert_rules.yaml is when CNCC Core Ingress Rate crosses 70 % of 1000 (Maximum ingress request rate)
Severity	Minor
Alert details provided	**Description**: 'CNCC Core Ingress traffic Rate is above the configured minor threshold i.e. 700 requests per second (current value is: {{ $value }})' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Traffic Rate is above 70 Percent of Max requests per second(1000)' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is above 70 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8001
Metric Used	oc_ingressgateway_http_requests_total
Resolution	The alert is cleared either when the total Ingress traffic rate falls below the minor threshold or when the total traffic rate crosses the major threshold, in which case the *CnccCoreTotalIngressTrafficRateAboveMajorThreshold* alert is raised. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. Steps: 1. Reassess why the CNCC Core is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

5.2.2 CnccCoreTotalIngressTrafficRateAboveMajorThreshold

Table 5-17 CnccCoreTotalIngressTrafficRateAboveMajorThreshold

Trigger Condition	The total CNCC Core Ingress Message rate has crossed the configured major threshold of 800 TPS. Default value of this alert trigger point in cncc_alert_rules.yaml is when CNCC Core Ingress Rate crosses 80 % of 1000 (Maximum ingress request rate)
Severity	Major

Table 5-17 (Cont.) CnccCoreTotalIngressTrafficRateAboveMajorThreshold

Alert details provided	**Description**: 'CNCC Core Ingress traffic Rate is above the configured major threshold i.e. 800 requests per second (current value is: {{ $value }})' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is above 80 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8001
Metric Used	oc_ingressgateway_http_requests_total
Resolution	The alert is cleared when the total Ingress Traffic ratefalls below the Major threshold or when the total traffic rate cross the Critical threshold, in which case the CnccCoreTotalIngressTrafficRate Above CriticalThreshold. **Note:** The threshold is configurable in the *alerts.yaml* file. Steps: 1. Reassess why the CNCC Core is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

5.2.3 CnccCoreTotalIngressTrafficRateAboveCriticalThreshold

Table 5-18 CnccCoreTotalIngressTrafficRateAboveCriticalThreshold

Trigger Condition	The total CNCC Core Ingress Message rate has crossed the configured critical threshold of 900TPS. Default value of this alert trigger point in cncc_alert_rules.yaml is when CNCC Core Ingress Rate crosses 90 % of 1000 (Maximum ingress request rate)
Severity	Critical
Alert details provided	**Description**: 'CNCC Core Ingress traffic Rate is above the configured critical threshold i.e. 900 requests per second (current value is: {{ $value }})' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is above 90 Percent of Max requests per second(1000)'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8001
Metric Used	oc_ingressgateway_http_requests_total

Table 5-18 (Cont.) CnccCoreTotalIngressTrafficRateAboveCriticalThreshold

Resolution	The alert is cleared when the Ingress Traffic rate falls below the Critical threshold. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. Steps: 1. Reassess why the CNCC IAM is receiving additional traffic. 2. If this is unexpected, contact My Oracle Support.

5.2.4 CnccCoreMemoryUsageCrossedMinorThreshold

Table 5-19 CnccCoreMemoryUsageCrossedMinorThreshold

Trigger Condition	A pod has reached the configured minor threshold(70%) of its memory resource limits.
Severity	Minor
Alert details provided	**Description**: 'CNCC Core Memory Usage for pod {{ $labels.pod }} has crossed the configured minor threshold (70%) (value={{ $value }}) of its limit.' **Summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 70% of its limit.'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8002
Metric Used	container_memory_usage_bytes kube_pod_container_resource_limits Note : This is a kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the memory utilization falls below the Minor Threshold or crosses the major threshold, in which case CnccCoreMemoryUsageCrossedMajorThreshold alert is raised. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. If guidance is required, contact My Oracle Support.

5.2.5 CnccCoreMemoryUsageCrossedMajorThreshold

Table 5-20 CnccCoreMemoryUsageCrossedMajorThreshold

Trigger Condition	A pod has reached the configured major threshold(80%) of its memory resource limits.
Severity	Major

Table 5-20 (Cont.) CnccCoreMemoryUsageCrossedMajorThreshold

Alert details provided	**Description**: 'CNCC Core Memory Usage for pod {{ $labels.pod }} has crossed the configured major threshold (80%) (value = {{ $value }}) of its limit.' **Summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 80% of its limit.'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8002
Metric Used	container_memory_usage_bytes kube_pod_container_resource_limits Note : This is a kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the memory utilization falls below the Major Threshold or crosses the critical threshold, in which case CnccCoreMemoryUsageCrossedCriticalThreshold alert is raised **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. If guidance is required, contact My Oracle Support.

5.2.6 CnccCoreMemoryUsageCrossedCriticalThreshold

Table 5-21 CnccCoreMemoryUsageCrossedCriticalThreshold

Trigger Condition	A pod has reached the configured critical threshold (90%) of its memory resource limits
Severity	Critical
Alert details provided	**Description**: 'CNCC Core Memory Usage for pod {{ $labels.pod }} has crossed the configured critical threshold (90%) (value = {{ $value }}) of its limit.' **Summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Memory Usage of pod exceeded 90% of its limit.'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8002
Metric Used	container_memory_usage_bytes kube_pod_container_resource_limits Note : This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the memory utilization falls below the Critical Threshold. **Note:** The threshold is configurable in the *occncc_alertrules_<version>.yaml* file. If guidance is required, contact My Oracle Support.

5.2.7 CnccCoreTransactionErrorRateAbove0.1Percent

Table 5-22 CnccCoreTransactionErrorRateAbove0.1Percent

Trigger Condition	The number of failed transactions is above 0.1 percent of the total transactions..
Severity	Warning
Alert details provided	**Description**:'CNCC Core transaction Error rate is above 0.1 Percent of Total Transactions (current value is {{ $value }})' **Summary**:'CNCC Core transaction Error rate is above 0.1 Percent of Total Transactions (current value is {{ $value }})'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8003
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert is cleared when the number of failed transactions are below 0.1% of the total transactions or when the number of failed transactions cross the 1% threshold in which case the *CnccCoreTransactionErrorRateAbove1Percent* is raised. The threshold is configurable in the *alerts.yaml* file. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.2.8 CnccCoreTransactionErrorRateAbove1Percent

Table 5-23 CnccCoreTransactionErrorRateAbove1Percent

Trigger Condition	The number of failed transactions is above 1 percent of the total transactions.
Severity	Warning
Alert details provided	**Description**: 'CNCC Core transaction Error rate is above 1 Percent of Total Transactions (current value is {{ $value }})' **Summary**:'CNCC Core transaction Error Rate detected above 1 Percent of Total Transactions'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8003
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert is cleared when the number of failed transactions are below 1% of the total transactions or when the number of failed transactions crosses the 10% threshold in which case the *CnccCoreTransactionErrorRateAbove10Percent* is raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required,contact My Oracle Support.

5.2.9 CnccCoreTransactionErrorRateAbove10Percent

Table 5-24 CnccCoreTransactionErrorRateAbove10Percent

Trigger Condition	The number of failed transactions is above 10 percent of the total transactions.
Severity	Minor
Alert details provided	**Description**: 'CNCC Core transaction Error rate is above 10 Percent of Total Transactions (current value is {{ $value }})' **summary**: 'CNCC Core ransaction Error Rate detected above 10 Percent of Total Transactions'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8003
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert is cleared when the number of failed transactions are below 10% of the total transactions or when the number of failed transactions crosses the 25% threshold in which case the *CnccCoreTransactionErrorRateAbove25Percent* is raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.2.10 CnccCoreTransactionErrorRateAbove25Percent

Table 5-25 CnccCoreTransactionErrorRateAbove25Percent

Trigger Condition	The number of failed transactions is above 25 percent of the total transactions.
Severity	Major
Alert details provided	**Description**: 'CNCC Core transaction Error Rate detected above 25 Percent of Total Transactions (current value is {{ $value }})' **Summary**: 'CNCC Core transaction Error Rate detected above 25 Percent of Total Transactions'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8003
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert is cleared when the number of failed transactions are below 25% of the total transactions or when the number of failed transactions crosses the 50% threshold in which case the *CnccCoreTransactionErrorRateAbove50Percent* is raised. Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.2.11 CnccCoreTransactionErrorRateAbove50Percent

Table 5-26 CnccCoreTransactionErrorRateAbove50Percent

Trigger Condition	The number of failed transactions is above 50 percent of the total transactions.
Severity	Critical
Alert details provided	**Description**: 'CNCC Core transaction Error Rate detected above 50 Percent of Total Transactions (current value is {{ $value }})' **Summary**: 'CNCC Core transaction Error Rate detected above 50 Percent of Total Transactions'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8003
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert is cleared when the number of failed transactions are below 50 percent of the total transactions Steps: 1. Check the Service specific metrics to understand the specific service request errors. 2. If guidance is required, contact My Oracle Support.

5.2.12 CnccCoreIngressGatewayServiceDown

Table 5-27 CnccCoreIngressGatewayServiceDown

Trigger Condition	Cncc Core Ingress Gateway service is down
Severity	Critical
Alert details provided	**Description**: 'CNCC Core Ingress-Gateway service InstanceIdentifier=~".*core_ingressgateway" is down' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Ingress-gateway service down' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }} : Ingress-gateway service down'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8004
Metric Used	'up' Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use a similar metric as exposed by the monitoring system.

Table 5-27 (Cont.) CnccCoreIngressGatewayServiceDown

Resolution	The alert is cleared when the cncc-core_ingressgateway service is available. Steps: 1. Check the orchestration logs of cncc-core_ingressgateway service and check for liveness or readiness probe failures. 2. Refer the application logs on Kibana and filter based on cncc-core_ingressgateway service names. Check for ERROR WARNING logs related to thread exceptions. 3. Depending on the failure reason, take the resolution steps. 4. In case the issue persists, contact My Oracle Support.

5.2.13 CnccCoreFailedLogin

Table 5-28 CnccCoreFailedLogin

Trigger Condition	The count of failed login attempts in CNCC-Core by a user goes above '3'
Severity	Warning
Alert details provided	**Description**:'{{ $value }} failed Login attempts have been detected in CNCC Core for user {{$labels.UserName}}, the configured threshold value is 3 failed login attempts for every 5 min' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: failed login attempts are more than the configured threshold value' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: failed login attempts are more than the configured threshold value'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8005
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert gets cleared when the total failed login attempts for a particular user go below the threshold value (default value is '3') in the last 5 min (default value is 5 m) **Note:** The threshold and time is configurable in the *alerts.yaml* file. If guidance is required, contact My Oracle Support.

5.2.14 CnccCoreUnauthorizedAccess

Table 5-29 CnccCoreUnauthorizedAccess

Trigger Condition	The count of failed login attempts in CNCC-Core by a user goes above '3'
Severity	Warning

Table 5-29 (Cont.) CnccCoreUnauthorizedAccess

Alert details provided	**Description**:'{{ $value }} Unauthorized Accesses have been detected in CNCC-Core for {{$labels.ResourceType}} for {{$labels.Method}} request. The configured threshold value is 3 for every 5 min' **For CNE 1.9.0 or later versions:** **summary**: 'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Unauthorized Access for CNCC-Core are more than threshold value' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Unauthorized Access for CNCC-Core are more than threshold value'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8006
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert gets cleared when the total failed login attempts for a particular user go below the threshold value (default value is '3') in the last 5 min (default value is 5 m) **Note:** The threshold and time is configurable in the alerts.yaml If guidance is required, contact My Oracle Support.

5.2.15 CnccCoreAccessTokenFailure

Table 5-30 CnccCoreAccessTokenFailure

Trigger Condition	If the count of failed token for CNCC-Core goes above configured value of '3'
Severity	Warning
Alert details provided	**Description**: 'CNCC Core Access Token Failure count is above the configured value i.e. 3 for every 5 min. Failed access token request count per second is (current value is: {{ $value }})' **For CNE 1.9.0 or later versions:** **summary:** 'namespace: {{$labels.namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . \| first \| value \| humanizeTimestamp }}{{ end }}: Access Token Failure count is above the configured threshold value' **For CNE 1.8.x or previous versions:** **summary**: 'namespace: {{$labels.kubernetes_namespace}}, podname: {{$labels.kubernetes_pod_name}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Access Token Failure count is above the configured threshold value'
OID used for SNMP Traps	1.3.6.1.4.1.323.5.3.51.1.2.8007
Metric Used	oc_ingressgateway_http_responses_total
Resolution	The alert gets cleared when the total failed tokens for a particular user go below the threshold value (default value is '3') in the last 5 min (default value is 5 m) **Note:** The threshold and time is configurable in the *alerts.yaml* file. If guidance is required, Contact My Oracle Support.